
Claude マニュアル

リリース 1.0.0

claude-manual.com

2026 年 10 月 06 日

目次

第 I 部	はじめに	1
第 1 章	Claude とは？— 全体像をつかもう	3
1.1	Claude は「安全性を重視した」AI アシスタント	3
1.2	Claude の製品ファミリー — どこからでも使える	3
1.3	あなたに合った Claude の選び方	5
1.4	料金プランの概要	6
1.5	次のステップ — ここから始めよう	7
1.6	もっと知りたい方向け: Claude を支える AI モデル	7
第 2 章	アカウント作成と初期設定	9
2.1	Claude を使い始める前に	9
2.2	アカウントの作成	10
2.3	ログインとホーム画面の確認	11
2.4	最初にやるべき設定	11
2.5	モバイルアプリのインストール (任意)	12
2.6	よくあるトラブルと解決策	13
2.7	次のステップ	14
第 3 章	Claude の料金プラン比較	15
3.1	プラン一覧比較表	15
3.2	Free プランでできること・できないこと	16
3.3	Pro プラン — 個人ユーザーに最も人気	16
3.4	Max プラン — ヘビーユーザー・開発者向け	17
3.5	Team プラン — チームでの導入に	18
3.6	Enterprise プラン — 大規模組織向け	19
3.7	おすすめの選び方フローチャート	20
3.8	よくある質問	21
3.9	次のステップ	22
第 4 章	はじめての会話 — Claude に話しかけてみよう	23
4.1	まず話しかけてみよう	23
4.2	Claude の回答を読む	24
4.3	会話を続ける	24
4.4	ファイルや画像を送る	25
4.5	うまく伝えるコツ	26
4.6	よくある疑問と注意点	27
4.7	次のステップ	28
第 5 章	Claude の料金を日本円で解説【2026 年最新】	29
5.1	Claude の料金一覧 (日本円換算)	29

5.2	無料プランでどこまで使える？	30
5.3	Pro プラン（月額約 3,000 円）の費用対効果	31
5.4	年払いと月払いの違い	31
5.5	支払い方法	32
5.6	学生・教育機関向けの料金	33
5.7	よくある質問	34
5.8	次のステップ	35
第 6 章	Claude Pro の制限と対処法 — 使い切った時にやるべきこと	37
6.1	Claude Pro でできること — まず価値を確認	37
6.2	Pro プランの利用制限 — 仕組みを正確に理解する	38
6.3	「上限に達しました」が出た時の対処法	39
6.4	利用量を節約する 5 つのコツ	40
6.5	Claude Code と利用枠の関係	41
6.6	Max プランに切り替えるべきタイミング	42
6.7	Free vs Pro vs Max 利用量比較	42
6.8	よくある質問	43
6.9	次のステップ	44
第 7 章	Claude のモデル比較・選び方 — Opus・Sonnet・Haiku どれを使う？	45
7.1	3 つのモデル早わかり表	45
7.2	Opus（オーパス）— じっくり考える最高性能モデル	45
7.3	Sonnet（ソネット）— 日常使いのバランス型	46
7.4	Haiku（ハイク）— 素早く答える軽量モデル	47
7.5	用途別おすすめ早見表	48
7.6	モデルの切り替え方	48
7.7	番外編: Mythos — セキュリティ研究向けモデル	49
7.8	よくある質問	49
7.9	次のステップ	50
第 8 章	安全な使い方・AI の限界	51
8.1	AI は万能ではない	51
8.2	ハルシネーション——もっともらしい嘘に注意	52
8.3	個人情報・機密情報の扱い	52
8.4	著作権・知的財産	53
8.5	バイアスと公平性——AI の回答には偏りがある	54
8.6	AI に任せてはいけないこと	55
8.7	Claude の安全設計について	55
8.8	安全に使うためのチェックリスト	55
8.9	関連リンク	56
第 9 章	Claude と ChatGPT の違いを徹底比較【2026 年最新】	57
9.1	結論：3 秒でわかる比較表	57
9.2	基本情報の比較	58
9.3	料金プランの比較	58
9.4	性能・得意分野の比較	59
9.5	ユースケース別おすすめ	61
9.6	併用のすすめ	62

9.7	よくある質問	63
9.8	まとめ	64
9.9	次のステップ	64
第 10 章	Claude と Gemini の違いを比較【2026 年最新】	65
10.1	結論 — 忙しい人向けの要約	65
10.2	基本情報の比較	65
10.3	料金プランの比較	66
10.4	性能・得意分野の比較	67
10.5	Google エコシステムとの統合	69
10.6	ユースケース別おすすめ	70
10.7	併用のすすめ	71
10.8	よくある質問	71
10.9	まとめ	72
10.10	次のステップ	73
第 11 章	AI チャット比較【2026 年版】 Claude ・ ChatGPT ・ Gemini ・ Copilot	75
11.1	結論：忙しい人向け一覧比較表	75
11.2	各サービスの概要	76
11.3	料金プランの横断比較	78
11.4	性能比較	79
11.5	ベンチマークとの付き合い方	81
11.6	用途別おすすめランキング	81
11.7	無料で始めるならどれ？	82
11.8	よくある質問	83
11.9	まとめ	84
11.10	次のステップ	84
第 II 部	製品ガイド	87
第 12 章	Claude Web (claude.ai) の使い方	89
12.1	Claude Web とは	89
12.2	基本の会話	90
12.3	Artifacts — 会話の横で「モノ」を作る	91
12.4	Projects — 仕事をまとめて管理する	91
12.5	Deep Research — 複雑な調査を Claude に任せる	92
12.6	Memory — Claude があなたを覚えてくれる	93
12.7	Connectors — 外部サービスと連携する	93
12.8	モデルの切り替え — 用途に合わせて選ぶ	94
12.9	次のステップ	95
第 13 章	Artifacts — 会話の横で「モノ」を作る	97
13.1	Artifacts とは	97
13.2	Artifacts の種類	98
13.3	Artifacts の基本操作	99
13.4	活用例：ビジネス文書の作成	100
13.5	活用例：データの可視化	101

13.6	活用例：ミニアプリ・ツールの作成	102
13.7	Artifacts カタログ — 他の人が作ったものを活用する	103
13.8	次のステップ	104
第 14 章	Deep Research — 複雑な調査を Claude に任せる	105
14.1	Deep Research とは	105
14.2	使い方	106
14.3	向いているテーマ・向いていないテーマ	107
14.4	良い調査依頼の書き方	108
14.5	レポートの読み方と活用	109
14.6	利用上限と料金	110
14.7	次のステップ	111
第 15 章	Memory — Claude があなたを覚えてくれる	113
15.1	Memory とは	113
15.2	Memory に覚えてもらえること	113
15.3	Memory の使い方	114
15.4	Chat Search（会話履歴の検索）	115
15.5	Memory Import（他サービスからの移行）	116
15.6	プライバシーと安全性	117
15.7	効果的な Memory の活用法	118
15.8	次のステップ	119
第 16 章	Projects — 会話とナレッジをまとめて管理する	121
16.1	Projects とは	121
16.2	プロジェクトの作り方	122
16.3	カスタム指示を設定する	122
16.4	ナレッジを活用する	123
16.5	活用例	125
16.6	チームで共有する（Team / Enterprise）	126
16.7	プランごとの違い	128
16.8	次のステップ	128
第 17 章	Claude Code をはじめよう	129
17.1	Claude Code とは	129
17.2	前提条件	130
17.3	インストール	131
17.4	認証（ログイン）	132
17.5	CLAUDE.md の作成	133
17.6	基本的な使い方	134
17.7	主要なスラッシュコマンド	135
17.8	実際に試してみる	136
17.9	次のステップ	136
第 18 章	非エンジニア向け Claude Code ガイド	137
18.1	なぜ非エンジニアにも Claude Code が役立つのか	137
18.2	ターミナルは怖くない — 3 分でわかる基礎知識	138
18.3	インストール手順（丁寧版）	138

18.4	非エンジニアのための活用タスク 5 選	141
18.5	安全に使うためのポイント	143
18.6	よくある不安 Q&A	144
18.7	Cowork との使い分け	145
18.8	次のステップ	145
第 19 章	Claude Code 実践ワークフロー集	147
19.1	ワークフローの考え方	147
19.2	コードレビュー	148
19.3	バグ修正	149
19.4	リファクタリング	151
19.5	テスト生成	152
19.6	ドキュメント生成	153
19.7	新規プロジェクトの立ち上げ	155
19.8	非エンジニア向けワークフロー	156
19.9	次のステップ	158
第 20 章	Claude Code の IDE 統合 (VS Code / JetBrains)	159
20.1	IDE 統合の概要	159
20.2	VS Code 拡張機能	160
20.3	JetBrains プラグイン	162
20.4	日常の開発フロー	164
20.5	ターミナル版との使い分け	166
20.6	設定とカスタマイズ	166
20.7	トラブルシューティング	167
20.8	次のステップ	169
第 21 章	Claude Desktop アプリの使い方	171
21.1	Claude Desktop とは — Web 版との何が違うの？	171
21.2	インストール	172
21.3	Chat モード — Web 版と同じ感覚で使える	173
21.4	Cowork モード — Claude にタスクを丸ごと任せる	174
21.5	MCP 連携 — 外部ツールと繋げる	176
21.6	Chrome 拡張・Office アドインとの連携	177
21.7	よくある質問	177
21.8	次のステップ	178
第 22 章	Cowork 入門 — Claude にタスクを任せよう	179
22.1	Cowork とは — 「会話」ではなく「タスク実行」	179
22.2	動作環境 — デスクトップアプリが必要	180
22.3	セットアップ — 3 ステップで準備完了	180
22.4	最初のタスクを実行する	181
22.5	タスクの書き方 — 良い指示の出し方	182
22.6	進行状況の確認と方向転換	183
22.7	スケジュールタスク — 繰り返しの仕事を自動化する	184
22.8	外部サービス連携 — Slack・Notion・Gmail と繋げる	185
22.9	セキュリティと安全性	186
22.10	次のステップ	186

第 23 章 Cowork プラグイン — 機能を拡張しよう	187
23.1 プラグインとは — Cowork の「連携ブリッジ」	187
23.2 公式プラグインの導入手順	188
23.3 プラグインの管理	189
23.4 主要プラグイン紹介	190
23.5 複数プラグインを組み合わせる	193
23.6 カスタムプラグインの作成（概要）	194
23.7 トラブルシューティング	194
23.8 次のステップ	195
第 24 章 Claude in Chrome — ブラウザ上で Claude に仕事を任せる	197
24.1 Claude in Chrome とは	197
24.2 インストールとセットアップ	198
24.3 基本的な使い方	199
24.4 活用例	200
24.5 Claude Code との連携（開発者向け）	203
24.6 Cowork との使い分け	203
24.7 安全に使うために	204
24.8 プランごとの違い	206
24.9 次のステップ	206
第 III 部 活用ガイド	207
第 25 章 ビジネス活用ユースケース	209
25.1 ビジネスで Claude を使うメリット	209
25.2 メール・文書作成	210
25.3 データ分析・レポート	210
25.4 リサーチ・情報収集	211
25.5 プレゼン資料の作成	212
25.6 翻訳・多言語対応	213
25.7 経理・財務	214
25.8 営業・顧客対応	216
25.9 企業での導入事例	217
25.10 機密情報の取り扱いについて	218
25.11 次のステップ	218
第 26 章 エンジニアリング活用ユースケース	219
26.1 エンジニアリングで Claude を使うメリット	219
26.2 コーディング支援	220
26.3 デバッグ	221
26.4 コードレビュー	221
26.5 設計・アーキテクチャ	222
26.6 テスト	223
26.7 ドキュメント	224
26.8 学習・キャッチアップ	225
26.9 セキュリティ上の注意	226
26.10 実践 Tips	227

26.11 次のステップ	228
第 27 章 マーケティング活用ユースケース	229
27.1 マーケティングで Claude を使うメリット	229
27.2 コンテンツ制作 — ブログ・ニュースレター・プレスリリース	230
27.3 SNS 運用 — 投稿文・ハッシュタグ・スケジュール	231
27.4 SEO — キーワードリサーチと構成案	232
27.5 市場分析・競合調査 — Deep Research で深掘りする	233
27.6 広告コピー — A/B テスト用バリエーションを一気に作る	234
27.7 メールマーケティング — セグメント別の文面を効率化	235
27.8 レポート作成 — KPI レポートと経営層向けサマリー	236
27.9 AI 生成コンテンツを使う際の注意点	237
27.10 次のステップ	238
第 28 章 プロンプトの基本原則	239
28.1 なぜ「指示の書き方」が重要なのか	239
28.2 原則 1: 明確に伝える	240
28.3 原則 2: 具体的に伝える	241
28.4 原則 3: 文脈を提供する	243
28.5 3 原則をまとめて使う	245
28.6 実践ワークショップ	246
28.7 よくある質問	247
28.8 次のステップ	248
第 29 章 プロンプトパターン集	249
29.1 パターンの読み方	249
29.2 パターン 1: Chain of Thought (段階的思考)	249
29.3 パターン 2: Few-shot (具体例の提示)	251
29.4 パターン 3: XML タグ構造化	253
29.5 パターン 4: ペルソナ / ロール設定	255
29.6 パターン 5: 分割と征服	257
29.7 パターン 6: 自己検証	258
29.8 パターン 7: マルチターン戦略	260
29.9 パターンの組み合わせ	261
29.10 次のステップ	263
第 30 章 Claude を AI 秘書にする	265
30.1 AI 秘書に任せられる 5 つの業務	265
30.2 レベル別の始め方	266
30.3 実践者の声	267
30.4 AI 秘書の限界と使い分け	268
30.5 もっと詳しく	268
30.6 よくある質問	269
30.7 関連記事	269
第 31 章 メール処理を Claude で効率化する	271
31.1 なぜメール処理はこんなに時間がかかるのか	271
31.2 Level 1: コピペで始めるメール下書き	271

31.3	Level 2: Claude Desktop コネクタで Gmail 接続	274
31.4	Level 3: MCP で本格的なメール管理	275
31.5	「人間が書くべきメール」の線引き	276
31.6	コピペで使えるプロンプト集	277
31.7	次のステップ	278
第 32 章	朝のタスク整理を自動化する	279
32.1	Level 1: claude.ai でコピペ・ブリーフィング	279
32.2	Level 2: Claude Desktop のコネクタで自動収集	281
32.3	Level 3: 「おはよう」で全自動ブリーフィング	282
32.4	週次レビューの自動化	284
32.5	実践者のワークフロー比較	284
32.6	今日から始めるための最初の一步	285
32.7	次のステップ	285
第 33 章	日報・週報・報告書を Claude で作る	287
33.1	日報・週報が続かない本当の理由	287
33.2	Level 1: コピペで日報を作る	287
33.3	Level 2: カレンダー・チャットから自動生成する	289
33.4	報告書・提案書の下書きを作る	291
33.5	「振り返り」は人間がやる——これが唯一の黄金律	293
33.6	コピペで使えるプロンプト集	295
33.7	次のステップ	297
第 34 章	AI 秘書を本格化する — MCP 連携と複数ツール統合	299
34.1	Level 1・2 の限界と Level 3 で変わる事	299
34.2	MCP とは何か	300
34.3	実践者のアーキテクチャ	300
34.4	よく使われる MCP サーバー	302
34.5	段階的自律性の設計	303
34.6	ポリシーファイルで AI の行動を制御する	304
34.7	セキュリティとプライバシー	305
34.8	始め方のロードマップ	306
34.9	まとめ: AI 秘書の完成形	307
34.10	次のステップ	307
第 35 章	Claude で PDF・ファイルを活用する	309
35.1	Claude にファイルを渡すことができる事	309
35.2	ファイルをアップロードする手順	309
35.3	対応形式と制限	310
35.4	Claude が PDF を読む仕組み	310
35.5	PDF 活用の基本パターン	311
35.6	うまく読み取れないときは	311
35.7	Projects で繰り返し使う	312
35.8	もっと詳しく	312
35.9	よくある質問	313
35.10	次のステップ	313

第 36 章 Claude で PDF を要約する方法とプロンプト例	315
36.1 要約は「短くすること」ではない	315
36.2 シーン別プロンプトテンプレート	315
36.3 長い PDF (100 ページ超) を要約するコツ	318
36.4 要約の精度を上げるテクニック	319
36.5 つまづきポイント	320
36.6 次のステップ	321
第 37 章 Claude で PDF が読み込めないときの対処法	323
37.1 まず確認するチェックリスト	323
37.2 ケース別の対処法	324
37.3 それでも解決しないときは	328
37.4 次のステップ	329
第 38 章 Claude でファイルを出力・生成する方法	331
38.1 Artifacts (アーティファクト) とは	331
38.2 Claude が出力できる形式	331
38.3 実用例 1: レポート・提案書を PDF として保存する	332
38.4 実用例 2: データを CSV で出力して Excel で開く	333
38.5 実用例 3: SVG でグラフや図を画像として保存する	333
38.6 実用例 4: Word・PowerPoint を直接生成する (有料プラン)	333
38.7 きれいな PDF を作るプロンプトのコツ	334
38.8 無料プランでの代替手段	335
38.9 Artifacts が表示されない・ダウンロードできないときは	335
38.10 次のステップ	335
第 39 章 Claude で PDF を翻訳する方法	337
39.1 基本の翻訳手順	337
39.2 分野別プロンプトテンプレート	337
39.3 翻訳精度を上げるテクニック	339
39.4 Google 翻訳・DeepL との使い分け	341
39.5 よくあるつまづきポイント	341
39.6 次のステップ	342
第 IV 部 上級テクニック	343
第 40 章 MCP ガイド — Claude に外部ツールをつなげる	345
40.1 MCP とは — アーキテクチャの全体像	345
40.2 既成サーバーの利用 — すぐに使えるサーバー一覧	346
40.3 Claude Desktop での設定	347
40.4 Claude Code での設定	350
40.5 自作サーバー入門	351
40.6 デバッグとトラブルシューティング	354
40.7 セキュリティの考慮事項	355
40.8 次のステップ	356
第 41 章 API 入門 — Claude をアプリに組み込む	357
41.1 API の概要	357

41.2	クイックスタート	358
41.3	Messages API の基本	360
41.4	Function Calling (ツール使用)	363
41.5	ストリーミング	366
41.6	Extended Thinking (拡張思考)	368
41.7	バッチ処理	369
41.8	コスト最適化	371
41.9	次のステップ	373
AI・Claude 用語集		375
	基本用語	375
	Claude 固有の用語	376
	モデル関連	379
	プロンプト関連	380
	API 関連	382
	セキュリティ・安全性関連	384
	次のステップ	385

この文書について

本書は、Web サイト「Claude マニュアル」の日本語版を、個人で読むために PDF 向けに組み直したものです。内容は変換元のままで、加筆や要約はしていません。著作権は変換元のサイトにあります。

項目	内容
変換元	https://claude-manual.com/ja
取得日	2026-10-06
収録範囲	日本語版の全 42 ページ。並び順はサイトのナビゲーションと同じです。

Web 版との違いは次のとおりです。

Web 版の要素	本書での扱い
セクション（はじめに、製品ガイドなど）	部にしています。
サイドバーの小見出し（Claude Web、AI 秘書など）	章立てには出していません。章の並び順で区別してください。
タブ、折りたたみ	見出し語と字下げで、すべての内容を順に掲載しています。
サイト内のリンク	文書内の参照にし、ページ番号を添えています。
用語集	部に入れず、巻末に章番号なしで収録しています。
絵文字	PDF のフォントに字形が無いため、✓、×、[赤] などに置き換えています。
太字にならず ** が残っている箇所	太字にしています（5 か所）。

サイト内のリンクのうち、リンク先のページがサイトに存在しないものが 3 か所あります。これらは元の URL のまま残しています。

第I部

はじめに

第1章 Claude とは？— 全体像をつかもう

「AI が便利らしいけど、何から始めればいいのか?」「ChatGPT は聞いたことあるけど、Claude って何が違うの?」

そんな疑問を持つ方のためのページです。このページを読めば、Claude の全体像がつかめて、**自分がまず何を試せばいいか**がわかります。

1.1 Claude は「安全性を重視した」AI アシスタント

Claude (クロード) は、AI 企業 Anthropic (アンソロピック) が開発した AI アシスタントです。文章の作成、質問への回答、データの分析、プログラミングの補助など、幅広い知的作業をこなします。

ChatGPT (OpenAI 社) や Gemini (Google 社) と同じ「大規模言語モデル (LLM)」と呼ばれる技術がベースですが、Claude には以下のような特徴があります。

- **安全性へのこだわり** — Anthropic は「AI の安全な開発」を掲げて設立された企業です。Claude は有害な回答を避けるよう慎重に設計されています
- **長い文章の理解が得意** — 最大 100 万トークン (日本語で約 50 万文字、文庫本 5~6 冊分) の文脈を一度に処理できます
- **自然な日本語対応** — 日本語での会話や文章作成が自然で、ビジネス文書の作成にも適しています
- **作業の代行ができる** — ただ質問に答えるだけでなく、ファイルの編集やコードの実行など、実際の作業を代行する「エージェント」としても使えます

つまづきポイント: 「LLM」「トークン」などの用語が出てきましたが、覚える必要はありません。大事なのは「Claude は賢い AI アシスタントで、いろいろなことを手伝ってくれる」ということです。

1.2 Claude の製品ファミリー — どこからでも使える

Claude はひとつのアプリではなく、**用途や環境に合わせた複数の製品**として提供されています。「Claude」という同じ AI の頭脳を、さまざまな形で使えると考えてください。

製品	どこで使う?	こんな人向け
Claude Web (claude.ai)	ブラウザ	今すぐ試したい人。インストール不要で一番手軽
Claude Desktop	Mac / Windows アプリ	日常的に使いたい人。PC の画面やファイルと連携できる

次のページに続く

表 1 – 前のページからの続き

製品	どこで使う？	こんな人向け
Claude モバイル	iPhone / Android	移動中にも使いたい人。スマホから手軽に質問・指示
Claude Code	ターミナル / IDE	エンジニア・開発者向け。コードの作成・編集を AI が代行

1.2.1 Claude Web (claude.ai) — まずはここから

ブラウザで claude.ai にアクセスするだけで使えます。アカウントを作れば無料プランですぐに始められます。

主な機能:

- **チャット** — 質問や相談をテキストで入力すると、Claude が回答します
- **ファイルのアップロード** — PDF や画像を読み込ませて内容を分析してもらえます
- **Artifacts (アーティファクト)** — Claude が作成した文書やコードを、その場でプレビューできる機能です。例: 「会議の議事録をまとめて」と頼むと、整形されたドキュメントがその場で表示され、コピーやダウンロードできます
- **Projects (プロジェクト)** — よく使う資料やルールをまとめておき、毎回の会話で自動的に参照させることができます。例: 自社の製品カタログを登録しておけば、「新商品の紹介メールを書いて」と頼むだけで正確な情報に基づいた文章が生成されます
- **Deep Research** — 複雑な調査を Claude に任せて、情報を集めてレポートにまとめてもらえます。例: 「国内 SaaS 市場の最新トレンドを調べて」と依頼すると、複数の情報源を横断して調査レポートを作成してくれます
- **Connectors (コネクターズ)** — Microsoft 365 (Outlook・OneDrive・SharePoint) や Slack、GitHub など 150 以上の外部サービスと連携できます。無料プランでも利用可能で、たとえば「今日届いたメールを要約して」のように、いつも使っているサービスの情報を Claude に直接渡せます

1.2.2 Claude Desktop — PC と深く連携

Mac や Windows にインストールして使うデスクトップアプリです。Web 版と同じチャット機能に加えて、PC との連携が強化されています。

- **Cowork (コワーク) モード** — Claude がバックグラウンドで作業を続けてくれる機能です。たとえば「この 30 ページの PDF を要約して」と指示して、自分は別の仕事に取りかかる——そんな使い方ができます
- **外部サービス連携 (MCP)** — MCP (Model Context Protocol) という仕組みを使って、Google Drive や Slack などの外部サービスと Claude を接続できます。たとえば Google Drive 上のファイルを Claude に直接読み込ませる、といったことが可能になります

1.2.3 Claude モバイル — いつでもどこでも

iPhone / Android アプリで、外出先でも Claude に相談できます。音声入力にも対応しているので、歩きながらアイデアを整理するといった使い方も可能です。

1.2.4 Claude Code — エンジニアの強力な相棒

ターミナル（コマンドライン）や VS Code などの開発環境で動く、エンジニア向けの AI アシスタントです。自然な言葉で指示するだけで、コードの作成・編集、テストの実行、Git 操作などを自動で行ってくれます。

つまづきポイント:「エンジニアじゃないけど Claude Code って使えるの?」と思った方。Claude Code はプログラミングの知識が必要なので、まずは **Claude Web** や **Claude Desktop** から始めるのがおすすめです。このサイトでは非エンジニアの方が Claude Code にチャレンジするための記事も用意しています。

1.3 あなたに合った Claude の選び方

「製品がたくさんあって迷う...」という方のために、やりたいこと別におすすめをまとめました。

1.3.1 「まず AI を試してみたい」方

おすすめ: Claude Web (claude.ai)

アカウントを作ってブラウザでアクセスするだけ。インストールも料金も不要です。まずは以下のようことを試してみましょう。

- 「来週の会議のアジェンダを作って」
- 「この文章を敬語に直して」
- 「〇〇について簡単に説明して」

1.3.2 事務・バックオフィスの方

おすすめ: Claude Web → Claude Desktop

日々の文書作成、データ整理、メールの下書きに活躍します。慣れてきたら Desktop アプリの Cowork モードで、レポート作成などの大きなタスクを任せてみましょう。

- 議事録の要約、報告書の作成
- Excel データの分析・グラフ化
- マニュアルや手順書の作成

1.3.3 営業・マーケティングの方

おすすめ: Claude Web + Claude モバイル

提案書の作成はデスクの Claude Web で、移動中のアイデア整理はモバイルアプリで。Projects 機能に商品情報や過去の提案書をまとめておくと、一貫性のある資料を素早く作れます。

- 提案書・企画書のドラフト
- 競合分析レポートの作成
- 顧客向けメールの下書き

1.3.4 経営・マネジメント層の方

おすすめ: Claude Web (Deep Research) → Claude Desktop (Cowork)

市場調査やレポート分析には Deep Research が便利です。大量の情報を Claude がまとめてくれるので、意思決定に集中できます。チーム導入には Team プランがおすすめです。

- 市場調査・競合分析
- 財務レポートの読み解き
- 戦略ドキュメントのレビュー

1.3.5 エンジニア・開発者の方

おすすめ: Claude Code

ターミナルや IDE で直接使える Claude Code が最適です。コードの作成・レビュー・デバッグ・リファクタリングを自然言語で指示できます。Git 操作やテスト実行も任せられます。

- コードの新規作成・修正
- バグの調査と修正
- プルリクエストの作成

1.4 料金プランの概要

Claude には用途や使用量に合わせた複数のプランがあります。

注意: 料金は変更される場合があります。最新の情報は [Anthropic 公式の料金ページ](#) で確認してください。

プラン	月額料金	ポイント
Free	無料	基本機能が使える。まず試すならこれ。メッセージ数に制限あり
Pro	\$20/月	Free の 5 倍の利用量。Claude Code や Cowork も利用可能

次のページに続く

表 2 – 前のページからの続き

プラン	月額料金	ポイント
Max 5x	\$100/月	Pro の 5 倍の利用量。ヘビーユーザー向け
Max 20x	\$200/月	Pro の 20 倍の利用量。最優先アクセス
Team	\$25/月～（1 人あたり）	チームでの共同利用向け。管理機能・メンバー管理つき（5 人～）
Enterprise	要問い合わせ	大規模組織向け。セキュリティ・管理機能が充実

まず無料プランで試して、もっと使いたくなったら Pro プランに切り替えるのがおすすめの流れです。Pro プランなら月額約 3,000 円（\$20）で、日常業務をカバーできる十分な利用量が確保できます。

つまづきポイント：「Free プランだとメッセージ上限がすぐ来てしまう」という声をよく聞きます。1 日あたりの利用量に制限があるため、本格的に使うなら Pro プラン以上をおすすめします。

料金プランの詳しい比較は「[Claude の料金プラン比較](#)（ページ 15）」をご覧ください。

1.5 次のステップ — ここから始めよう

あなたの状況に合わせて、次に読む記事を選んでください。

Claude Web を使ってみたい方（全員におすすめ）：

- [Claude Web \(claude.ai\) の使い方](#)（ページ 89）— ブラウザでの基本操作を学ぶ

Claude Desktop を使ってみたい方：

- [Claude Desktop アプリの使い方](#)（ページ 171）— デスクトップアプリのインストールと活用法

Claude Code を使ってみたい方（エンジニア向け）：

- [Claude Code をはじめよう](#)（ページ 129）— インストールから最初の一步まで

AI への指示のコツを知りたい方：

- [プロンプトの基礎](#)（ページ 239）— AI にうまく伝えるための基本を学ぶ

料金プランを詳しく知りたい方：

- [Claude の料金プラン比較](#)（ページ 15）— 各プランの違いを徹底比較

1.6 もっと知りたい方向け: Claude を支える AI モデル

Claude の「頭脳」にあたる AI モデルについて（クリックで展開）

Claude の AI モデルは、用途に応じて 3 つのクラスがあります。普段は意識する必要はありませんが、知っておくと便利です。

モデル	特徴	得意なこと
Opus（オーパス）	最も高性能	複雑な分析、高度な推論、長時間の作業

次のページに続く

表 3 – 前のページからの続き

モデル	特徴	得意なこと
Sonnet (ソネット)	バランス型	日常的なタスク全般。速度と性能のバランスが良い
Haiku (ハイク)	最速・低コスト	シンプルな質問、素早い応答が必要な場面

Free プランでは Sonnet が、Pro 以上のプランでは Opus も含めたすべてのモデルが利用できます。通常は Claude が自動的に最適なモデルを選んでくれるので、モデル選びで悩む必要はありません。詳しくは「[モデル比較・選び方](#) (ページ 45)」をご覧ください。

第2章 アカウント作成と初期設定

「Claude を試してみたいけど、登録がめんどくさそう...」「料金がかかるのかな？」

そんな心配は無用です。Claude は**無料で始められ**、登録もメールアドレスさえあれば5分もかかりません。このページでは、アカウントの作り方から最初の設定まで、画面の操作を1ステップずつ丁寧に解説します。

2.1 Claude を使い始める前に

まず、必要なものを確認しましょう。

2.1.1 必要なもの

必要なもの	説明
メールアドレス	フリーメール（Gmail など）でも OK
または Google アカウント	持っていれば最も簡単に登録できる（推奨）
スマートフォン	登録時に SMS（ショートメッセージ）で本人確認が必要
ブラウザ	Chrome、Safari、Edge など。最新版を推奨

2.1.2 対応デバイス・OS

デバイス	対応状況
Windows PC（Chrome/Edge）	対応
Mac（Chrome/Safari）	対応
iPhone / iPad	ブラウザ・専用アプリどちらも対応
Android	ブラウザ・専用アプリどちらも対応

料金について: Claude は**無料プラン（Free）**で始められます。クレジットカードの登録は不要です。無料プランでも基本的なチャット機能はすべて使えます。もっと使いたくなったときに、有料プランを検討すれば OK です。プランの違いが気になる方は「[Claude の料金プラン比較](#)（ページ 15）」をご覧ください。

2.2 アカウントの作成

claude.ai にアクセスします。以下の 3 通りの方法でアカウントを作れます。**Google アカウントを持っている方は「Google で登録」が最も簡単でおすすめです。**

2.2.1 方法 1: Google アカウントで登録（最も簡単・推奨）

1. 「**Continue with Google**（Google で続ける）」ボタンをクリックします
2. Google アカウントの選択画面が出たら、使いたいアカウントを選びます
3. Claude の利用規約と個人情報の取り扱いについての確認画面が出るので、内容を確認して同意します
4. 名前の入力画面が出たら、Claude との会話で使う名前を入力します（ニックネームでも OK）
5. 電話番号の確認画面が出ます。国を「Japan (+81)」に設定し、携帯電話番号を入力します（例: 09012345678）
6. SMS で 6 桁の確認コードが届くので、入力欄に入力します
7. 登録完了。Claude のホーム画面が表示されます

つまづきポイント: Google アカウントの選択画面でうまくいかない場合は、ブラウザを再起動してもう一度試してみてください。プライベートウィンドウ（シークレットモード）を使っている場合は、通常のウィンドウでお試ください。

2.2.2 方法 2: Apple アカウントで登録

Apple ID をお持ちの方は、「**Continue with Apple**（Apple で続ける）」ボタンから登録できます。Face ID やパスコードで認証が完了し、非常に簡単です。

注意: Apple での登録は、iOS / macOS の Safari または Claude のスマホアプリから行うのが確実です。他のブラウザでは表示されないことがあります。

2.2.3 方法 3: メールアドレスで登録

1. メールアドレス入力欄に、使いたいメールアドレスを入力します
2. 「**メールで続ける**」をクリックします
3. 入力したメールアドレスに確認メールが届きます。メール内の「**確認コードを入力**」またはリンクをクリックします
4. 届いた 6 桁の確認コードを入力します
5. 電話番号の確認画面が出ます。国を「Japan (+81)」に設定し、携帯電話番号を入力します
6. SMS で届いた 6 桁の確認コードを入力します
7. 名前を入力して、登録完了です

つまづきポイント: 確認メールが届かない場合、まず**迷惑メールフォルダ**を確認してください。それでも届かない場合は、メールアドレスの入力ミスの可能性があります。登録をやり直してみましょう。

2.3 ログインとホーム画面の確認

登録が完了すると、Claude のホーム画面（チャット画面）が表示されます。

画面の各パーツの役割を説明します。

2.3.1 左側のサイドバー

- **新しいチャット** — クリックすると新しい会話を始められます
- **チャット履歴** — 過去の会話が一覧で表示されます。クリックすると再度開けます
- **プロジェクト** — よく使う資料や設定をまとめておく場所です（後から設定できます）

2.3.2 中央の入力欄（メイン画面）

- **テキスト入力欄** — ここに質問や依頼を入力します。Enter キーで送信、Shift+Enter で改行できます
- **クリップマークのアイコン** — PDF やテキストファイル、画像などを添付できます
- **モデル選択** — 使用する AI モデルを選べます。通常はそのまま OK です

つまづきポイント: 「モデルって何を選べばいいの？」という方は、デフォルトの設定（Claude Sonnet）のままで始めましょう。モデルを変える必要が出てきたときに選び直せば OK です。

2.4 最初にやるべき設定

アカウントができたなら、使い始める前にいくつかの設定を確認しておきましょう。設定画面は、右上のアイコン（プロフィール）→「設定」から開けます。

2.4.1 1. 表示言語の確認

Claude は最初から日本語で使えます。入力欄に日本語で話しかければ、日本語で返答してくれます。もし英語表示になっている場合は、設定画面の「言語」から「日本語」に変更できます。

2.4.2 2. Memory（メモリ）機能の確認

Memory（メモリ） とは、Claude があなたの名前や好み、よく使う表現などを覚えておく機能です。設定しておくと毎回説明しなくて済むので、非常に便利です。

設定手順:

1. 設定画面を開く
2. 「機能」タブをクリック
3. 「メモリ」のスイッチがオンになっているか確認する

Memory 機能は 2026 年 3 月から無料プランでも利用できるようになりました。

つまづきポイント:「自分のことを覚えてほしくない」という場合は、Memory をオフにしても Claude は使えます。ただし、毎回のやりとりで背景情報を伝える必要があります。また、チャット履歴を削除してもメモリは削除されません。メモリも削除したい場合は、設定の「機能」→「メモリ」→「すべてのメモリを削除」から操作してください。

2.4.3 3. Artifacts（アーティファクト）の確認

Artifacts（アーティファクト）とは、Claude が作った文書やコードをその場でプレビューできる機能です。たとえば「議事録を整形して」と頼むと、チャットの横に整形済み文書がすぐ表示され、コピーやダウンロードができます。

設定手順:

1. 設定画面の「**機能**」タブを開く
2. 「**Artifacts**」のスイッチがオンになっているか確認する

現在は多くの場合デフォルトでオンになっていますが、確認しておきましょう。

2.4.4 4. データプライバシーの設定（任意）

設定画面の「**プライバシー**」から「Claude の改善にご協力ください」をオフにすることができます。これをオフにすると、あなたのチャット内容が AI の学習データとして使用されなくなります。

仕事上の機密情報や個人情報を Claude に入力する可能性がある場合は、オフにしておくことをおすすめします。

2.5 モバイルアプリのインストール（任意）

スマートフォンでも Claude を使いたい方は、専用アプリをインストールしましょう。ブラウザからも使えますが、アプリのほうが操作しやすく、音声入力にも対応しています。

2.5.1 iPhone / iPad の場合

App Store で「**Claude**」を検索してインストールします。

1. App Store を開く
2. 検索欄に「Claude」と入力する
3. Anthropic が提供するアプリをインストールする
4. アプリを開き、先ほど作成したアカウントでログインする（「Continue with Google」または「メールで続ける」）

2.5.2 Android の場合

Google Play ストアで「**Claude**」を検索してインストールします。

1. Google Play ストアを開く

2. 検索欄に「Claude」と入力する
3. Anthropic が提供するアプリをインストールする
4. アプリを開き、先ほど作成したアカウントでログインする（「Continue with Google」または「メールで続ける」）

どちらの場合も： Web 版と同じアカウントでログインするので、チャット履歴が引き継がれます。スマートフォンと PC で同じ会話を続けることができます。

2.6 よくあるトラブルと解決策

2.6.1 確認メールが届かない

1. **迷惑メールフォルダを確認**する。自動的に振り分けられていることが多いです
2. **数分待つ**。メールの到着が遅れることがあります
3. **メールアドレスの入力ミス**がないか確認し、もう一度登録をやり直す
4. 会社のメールアドレスを使っている場合、**セキュリティフィルタで遮断**されている可能性があります。Gmail などの個人メールアドレスで試してみてください

2.6.2 SMS の確認コードが届かない

1. **国コードを正しく選択**しているか確認する（日本は「Japan (+81)」）
2. 電話番号を入力するとき、**先頭の 0 も含めて入力**する（例: 09012345678）
3. **数分待つ**。SMS の届きが遅れることがあります
4. 電波の状況が悪い場合は、別の場所や Wi-Fi 環境で試す

つまづきポイント：「国コード (+81)」を選択した場合でも、電話番号は「090...」と先頭の 0 を含めて入力します。+81 を選んで「90...」と入力するのは誤りです。

2.6.3 ログインできない

- **パスワードを忘れた**場合は、ログイン画面の「パスワードをお忘れですか？」からリセットできます
- **メールアドレスを複数持っている**場合、どのアドレスで登録したか確認しましょう
- Google でログインした場合は、パスワードではなく「**Continue with Google**」ボタンからログインしてください

2.6.4 VPN を使っている場合

会社や自宅で VPN を使用している場合、地域制限により Claude にアクセスできないことがあります。VPN をオフにしてから接続を試みてください。

2.6.5 「地域でご利用いただけません」というメッセージが出る

日本からのアクセスは問題なく利用できますが、海外出張中や海外在住の場合、一部の国や地域では Claude が利用できません。現地の利用規約を確認してください。

2.7 次のステップ

アカウントが作成できたら、さっそく Claude と会話してみましょう。

まず Claude を使ってみたい方:

- [Claude との最初の会話](#) (ページ 23) — 最初の会話の始め方と、効果的な質問の仕方を学ぶ

料金プランを詳しく知りたい方:

- [Claude の料金プラン比較](#) (ページ 15) — Free・Pro・Team の違いを徹底比較

Claude の全体像を知りたい方:

- [Claude とは？— 全体像をつかもう](#) (ページ 3) — どの製品を使うべきか迷ったらここへ

第3章 Claude の料金プラン比較

「Claude を使い始めたけど、有料プランにすべき?」「Pro と Max、どっちがいいの?」「チームで使うならどのプランがお得?」——こうした疑問を持っている方に向けて、Claude の全プランをわかりやすく比較します。

この記事を読めば、自分（またはチーム）にぴったりのプランが見つかります。

料金に関する注意: この記事の料金情報は **2026 年 3 月時点** のものです。最新の料金は [Anthropic 公式の料金ページ](#) で確認してください。

3.1 プラン一覧比較表

Claude には、個人向けの **Free・Pro・Max** と、組織向けの **Team・Enterprise** の 5 つのプランがあります。

	Free	Pro	Max 5x	Max 20x	Team	Enterprise
月額料金	無料	\$20/月	\$100/月	\$200/月	Standard: \$25/席/月、Premium: \$125/席/月	要問い合わせ
年払い料金	—	\$17/月相当（年 \$200 前払い）	年払いオプションなし	年払いオプションなし	年払い割引あり	年間契約
利用量の目安	5 時間あたり約 15~40 メッセージ	1 日約 100~300 メッセージ相当 ※ 1	Pro の 5 倍	Pro の 20 倍	シート種別により異なる ※ 2	組織ごとにカスタム
Claude Code（AI コーディングツール）	不可	利用可	利用可（大容量）	利用可（最大容量）	利用可	利用可
使えるモデル	Sonnet・Haiku	Opus・Sonnet・Haiku	Opus・Sonnet・Haiku	Opus・Sonnet・Haiku	Opus・Sonnet・Haiku	Opus・Sonnet・Haiku
プロジェクト機能	最大 5 つまで	利用可	利用可	利用可	利用可	利用可
Connectors（外部サービス連携）	利用可（150+コネクタ）	利用可	利用可	利用可	利用可	利用可
Web 検索	利用可	利用可	利用可	利用可	利用可	利用可
管理機能	—	—	—	—	あり	高度な管理機能

※ 1 利用量の目安について: メッセージ数はあくまで目安です。1 回の会話の長さやモデルの種類によって実際に送れる回数は変わります。長い文書を添付したり、Opus モデルを使ったりすると消費量が増えます。

※ 2 Team プランのシート種別: Team プランには「Standard シート」と「Premium シート」の 2 種類があります。Standard シートは Pro プランとほぼ同等の利用量、Premium シートは Pro の約 6.25 倍の利用量と Claude Code が使えるため、チーム内でシートを組み合わせることで利用量に幅が出ます。Claude Code の利用はシート種別により異なります（詳細は公式サイト参照）。

3.2 Free プランでできること・できないこと

まずは無料の Free プランで Claude を試してみましょう。Free プランでも Claude の基本的な能力は体験できます。

Free プランでは **Claude Sonnet（標準モデル）** と **Claude Haiku（軽量・高速モデル）** が使えます。Sonnet は日常的な質問や文章作成には十分な性能を持っています。より高性能な **Claude Opus** モデルを使いたい場合は、Pro プラン以上へのアップグレードが必要です。

3.2.1 できること

- **チャットでの会話**: Web、iOS、Android、デスクトップアプリから Claude と会話できます
- **文章の作成・編集**: メールの下書き、文章の要約、翻訳など基本的な文章作業
- **画像の分析**: 写真やスクリーンショットをアップロードして、内容を説明してもらえます
- **コードの生成**: 簡単なプログラムの作成やコードの説明を依頼できます
- **Web 検索**: Claude がインターネットを検索して最新情報を調べてくれます
- **Connectors（外部サービス連携）**: Microsoft 365（Outlook・OneDrive・SharePoint）、Slack、GitHub など 150 以上の外部サービスと連携できます

3.2.2 できないこと・制限

- **利用量に上限がある**: 5 時間あたり約 15~40 メッセージ程度（会話の長さや混雑状況で変動します）
- **ピーク時にアクセスが制限される**: 平日の忙しい時間帯は、応答が遅くなったりアクセスできない場合があります
- **Claude Code が使えない**: ターミナルで動く AI コーディングツール「Claude Code」は有料プラン限定です
- **プロジェクト機能は最大 5 つまで**: カスタム指示やファイルを保存するプロジェクト機能は最大 5 つまで利用できます。ただし、高度な RAG（検索拡張生成）機能は有料プラン限定です
- **最上位モデルへのアクセスが限定的**: Claude Opus など最も高性能なモデルは使えないか、ごく限定的です
- **会話データがモデル改善に使われる可能性がある**: Free プランでは、会話内容が AI の学習に利用されることがあります

つまづきポイント: 「Free プランで使っていたら急にメッセージが送れなくなった」という声をよく聞きます。これは利用上限に達したためです。5 時間ほど待つとリセットされますが、頻繁に上限に当たる場合は Pro プランへのアップグレードがおすすめです。

3.3 Pro プラン — 個人ユーザーに最も人気

月額 \$20（年払いなら月 \$17 相当・年 \$200 前払い）の Pro プランは、Claude を日常的に使う個人ユーザーに最も人気のプランです。

3.3.1 Pro プランで変わること

- **利用量が大幅に増える**: Free プランの約 5 倍のメッセージを送れます
- **Claude Opus が使える**: 最も高性能なモデル「Claude Opus 4.6」にアクセスできます。複雑な分析や高品質な文章作成に最適です
- **Claude Code が使える**: ターミナルで動く AI コーディングツールが利用可能になります
- **プロジェクト機能**: よく使う指示やファイルを保存して、効率的に作業できます
- **拡張思考 (Extended Thinking)**: 複雑な問題に対して、Claude がじっくり考えてから回答する機能が使えます
- **Google Workspace 連携**: Gmail やカレンダー、ドキュメントと連携して使えます
- **優先アクセス**: 混雑時でも安定して利用できます

3.3.2 こんな人におすすめ

- 仕事や学習で **毎日 Claude を使う** 人
- **Claude Code** を使ってプログラミングをしたい人
- **高品質な文章** を作成したい人 (Opus モデルを使いたい)
- Free プランの **利用制限に不満** を感じている人

3.4 Max プラン — ヘビーユーザー・開発者向け

「Pro プランでも利用上限に当たってしまう」という方には **Max プラン** があります。Max には 2 つのティア (段階) があります。

	Max 5x	Max 20x
月額	\$100	\$200
利用量 (Pro 比)	5 倍	20 倍
向いている人	Pro の上限に頻繁に当たる人	Claude を一日中使い続ける人

3.4.1 Pro プランとの違い

Max プランでは、Pro で使えるすべての機能に加えて以下の点が異なります：

- **大幅に多い利用量**: Max 5x なら Pro の 5 倍、Max 20x なら 20 倍の利用量が使えます
- **新機能への早期アクセス**: 新しいモデルや機能をいち早く試せます
- **優先サービス**: ピーク時でも最優先で処理されます
- **超過分の追加購入**: 上限に達しても、API レートで追加利用分を購入できます

3.4.2 Claude Code の利用量について

Claude Code の利用量は、Claude（Web 版）と **共通の利用枠** を消費します。つまり、Web でたくさん会話した後に Claude Code を使うと、すぐに上限に達してしまうことがあります。

Claude Code はチャットと比べて **トークン（AI が処理するデータ量の単位）の消費が多い** ため、コーディング作業を中心にする方は余裕のある Max プランが向いています。

- **Pro プラン**: 軽めのコーディング作業なら十分。1 日に数回 Claude Code を使う程度の方向け
- **Max 5x**: Claude Code を 1 日 4 時間以上使うような開発者向け。プロジェクト全体のリファクタリングなど大きな作業にも対応
- **Max 20x**: Claude Code を中心に一日中開発する方向け。大規模なコードベースの分析や大量のファイル編集もストレスなく実行可能

つまづきポイント: 「Pro プランで Claude Code を使い始めたら、あっという間に上限に達してしまった」という方が多いです。Claude Code は会話よりもトークン消費が激しいため、開発がメインの方は最初から Max 5x を検討するとよいでしょう。

3.5 Team プラン — チームでの導入に

複数人で Claude を使いたい組織向けの **Team プラン** は、共同作業機能と管理機能が充実しています。

3.5.1 料金体系（2026 年 3 月時点）

- **最低人数**: 5 名から（最大 150 席）
- **Standard シート**: 月払い \$25/席/月、年払い \$20/席/月（個人の Pro プランとほぼ同等の利用量）
- **Premium シート**: 月払い \$125/席/月、年払い \$100/席/月（Pro の約 6.25 倍の利用量と Claude Code を含む上位シート）
- Claude Code の利用はシート種別により異なります（詳細は公式サイト参照）

3.5.2 Team プランの特徴

- **チャットの共有**: チームメンバーとチャット内容を共有し、共同で作業できます
- **管理コンソール**: メンバーの追加・削除、権限設定を一元管理できます
- **SSO（シングルサインオン）**: 会社の認証システムと連携してログインを簡素化できます
- **一括請求**: チーム全体の利用料を一括で管理・支払いできます
- **ドメインキャプチャ**: 会社のメールアドレスで登録したユーザーを自動的にチームに招待できます
- **支出管理**: 組織全体や個人ごとに利用上限を設定できます

3.5.3 こんな組織におすすめ

- 5～150 名のチームで Claude を活用したい
- メンバー間で ナレッジを共有 しながら使いたい
- 請求や権限を一元管理 したい
- セキュリティポリシーとして SSO が必要

3.6 Enterprise プラン — 大規模組織向け

大規模な組織や、厳格なセキュリティ要件がある企業向けの **Enterprise プラン** です。

3.6.1 Enterprise プランの特徴

Team プランのすべての機能に加えて：

- **SCIM（自動ユーザー管理）**：社員の入退社に合わせてアカウントを自動で管理できます
- **監査ログ**：誰がいつ何をしたかを記録・確認できます
- **カスタムデータ保持ポリシー**：データの保存期間を組織のポリシーに合わせて設定できます
- **ロールベースのアクセス制御**：部門や役職に応じて細かく権限を設定できます
- **IP アドレス制限**：特定のネットワークからのみアクセスを許可できます
- **HIPAA 対応**：医療データを扱う場合の準拠にも対応可能です
- **拡張コンテキストウィンドウ**：大量のドキュメントやコードを一度に処理できます
- **カスタム契約**：利用量、価格、サポート体制を個別に交渉できます

3.6.2 料金

Enterprise プランの料金は公開されていません。組織の規模や要件に応じた個別見積もりとなります。年間契約が基本で、最低 20 席からの導入となります。

詳しくは [Anthropic の Enterprise ページ](#) からお問い合わせください。

3.6.3 Team vs Enterprise — 管理機能・セキュリティ比較

組織での導入を検討している方向けに、Team プランと Enterprise プランの管理機能・セキュリティ機能を比較します。

機能	Team	Enterprise
最大人数	150 席	無制限
SSO（シングルサインオン）	対応	対応
SCIM（自動ユーザー管理）	非対応	対応
監査ログ	非対応	対応

[次のページに続く](#)

表 3 – 前のページからの続き

機能	Team	Enterprise
ロールベースのアクセス制御	基本的な権限管理	部門・役職別の細かい設定
IP アドレス制限	非対応	対応
カスタムデータ保持ポリシー	非対応	対応
HIPAA 対応	非対応	対応
拡張コンテキストウィンドウ	非対応	対応
請求方法	クレジットカード / 年払い	請求書払い / 年間契約
サポート	標準サポート	専任サポート・SLA あり

Enterprise プランを選ぶべき条件:

- チームの人数が **150 名を超える**
- **監査ログ** が社内のセキュリティポリシーで必須
- **HIPAA 準拠**が必要（医療・ヘルスケア業界）
- 社員の入退社に連動した **自動アカウント管理（SCIM）** が必要
- **IP アドレス制限** でアクセス元を限定したい
- データの保存期間を **自社ポリシーに合わせてカスタマイズ** したい

上記に該当しない場合は、まず Team プランで始めて、必要に応じて Enterprise にアップグレードするのが効率的です。

3.7 おすすめの選び方フローチャート

「結局どのプランがいいの？」と迷ったら、以下の質問に順番に答えてみてください。

Q1. Claude を使うのはあなた一人ですか？ それともチームですか？

- チーム（5～150 名）で使う → Team プラン へ
- チーム（150 名超）、または監査ログ・HIPAA 対応が必要 → Enterprise プラン へ
- 一人で使う → Q2 へ

Q2. Claude にお金を払う予定はありますか？

- まずは無料で試したい → Free プラン で始めましょう。物足りなくなったらアップグレードできます
- お金を払ってもいい → Q3 へ

Q3. Claude Code（AI コーディングツール）を使いますか？

- 使わない、または少しだけ使う → Pro プラン（\$20/月）がベストです
- 毎日がっつり使う → Q4 へ

Q4. Claude Code をどのくらい使いますか？

- 1 日 2～3 時間程度 → まず Pro プラン で試して、上限に頻繁に当たるなら Max へ

- 1日4時間以上 → **Max 5x** (\$100/月) がおすすめ
- ほぼ一日中使っている → **Max 20x** (\$200/月) が最適

3.7.1 迷ったらまず Pro から

どのプランにすべきか迷ったら、**まず Pro プラン (\$20/月) から始める** のがおすすめです。理由は：

- 月払いなので **いつでも解約・変更できる**
- 使い足りなければ **Max にアップグレード** できる
- 逆に使わなければ **Free に戻す** こともできる
- Pro プランの利用量は **多くのユーザーにとって十分** です

3.8 よくある質問

3.8.1 途中でプラン変更できますか？

はい、いつでもプランを変更できます。アップグレードは即時反映され、ダウングレードは次の請求サイクルから適用されます。

3.8.2 年払いと月払いの違いは？

年払い (Annual) を選ぶと、月払いよりも割安になります。たとえば Pro プランは月払いで \$20/月ですが、年払いなら \$17/月相当 (年 \$200 前払い) になります。ただし年払いは途中解約しても返金されない点に注意してください。

3.8.3 Free プランから有料プランにアップグレードするとデータは引き継がれますか？

はい、これまでの会話履歴はそのまま引き継がれます。

3.8.4 Claude Code は別料金ですか？

いいえ、Claude Code の利用料は Pro・Max プランの利用枠に含まれています。追加料金はかかりませんが、Claude (Web 版) と利用枠を共有している点に注意してください。

3.8.5 Team プランで個人ごとに利用量の上限を設定できますか？

はい、Team プランの管理機能で、個人ごとやチーム全体の支出上限 (スPENDキャップ) を設定できます。

3.8.6 支払い方法は何が使えますか？

クレジットカード (Visa、Mastercard、American Express 等) での支払いが基本です。Enterprise プランでは請求書払いにも対応しています。

3.8.7 学生割引はありますか？

個人向けの学生割引はありませんが、教育機関向けの **Education Plan** が存在します。大学単位でのキャンパス全体アクセスが可能です。詳しくは [Anthropic 公式サイト](#) を確認してください。個人で使いたい場合は、まず Free プランで始めることをおすすめします。

3.9 次のステップ

- [Claude とは？](#) (ページ 3) — Claude の概要と基本的な特徴を知る
- [Claude Web 版の使い方ガイド](#) (ページ 89) — ブラウザから Claude を使い始める方法
- [プロンプトの基本](#) (ページ 239) — AI にうまく指示を出すテクニック

第4章 はじめての会話 — Claude に話しかけてみよう

アカウントの作成が完了しましたね。次は、いよいよ Claude と話してみよう。

「AI に何を話しかければいいの?」「変なことを聞いたら怒られないかな?」——そんな不安を感じる方は多いです。でも大丈夫です。Claude は気軽に話しかけられる AI アシスタントで、どんな質問でも真剣に向き合ってくれます。

このページでは、最初のメッセージを送るところから、回答を読む方法、会話を続けるコツまでを順に解説します。

4.1 まず話しかけてみよう

ブラウザで claude.ai を開くと、チャット入力欄が表示されています。ここにメッセージを入力して **Enter** キー（または送信ボタン）を押すだけで、Claude との会話が始まります。

4.1.1 何を聞けばいいかわからない方へ — コピペで使えるお試しプロンプト

はじめて使うとき、「どんなことを頼めばいいのか」迷いますよね。以下のメッセージをそのままコピーして試してみましょう。

ビジネス文書の作成

社内向けの業務改善提案書の書き方を教えてください。
構成の例も示してもらえると助かります。

メールの下書き

取引先に納品遅延をお詫びするメールの文章を書いてください。
丁寧なビジネス敬語で、200 字程度でお願いします。

情報をわかりやすく説明してもらう

「生成 AI」とは何ですか？
IT に詳しくない 50 代の上司に説明するつもりで、やさしく教えてください。

翻訳

以下の英文を自然な日本語に翻訳してください。

"We are pleased to inform you that your application has been approved."

アイデア出し

新入社員向けのウェルカムランチの会場選びのポイントを 5 つ挙げてください。
予算は 1 人 3,000 円程度です。

つまづきポイント: 「何を聞けばいいかわからない」という状態のまま放置してしまう方が多いです。まずは上のプロンプトのどれかを試してみてください。1 回試せば「こういう感じか」とすぐ理解できます。

4.2 Claude の回答を読む

メッセージを送ると、数秒で Claude の回答が表示されます。回答は質問の内容に応じて、さまざまな形式で返ってきます。

4.2.1 回答の形式を理解する

Claude の回答によく使われる書式を知っておくと、素早く内容をつかめます。

書式	意味
太字	特に重要なポイント
箇条書き（・や-）	複数の項目を整理したリスト
番号付きリスト（1.2.3.）	手順や優先順位がある内容
コードブロック	プログラムコードや入力すべき文字列
見出し（大きな文字）	長い回答の章区切り

4.2.2 回答が長すぎる・短すぎると感じたら

Claude の回答が詳しくすぎて読みきれない場合や、逆に短すぎて物足りない場合は、次のように指示できます。

- 「もっと短くまとめて」
- 「要点を 3 行で教えて」
- 「もっと詳しく説明して」
- 「初心者向けにかみ砕いて説明して」

同じ会話の中で続けて指示するだけで OK です。次のセクションで詳しく説明します。

4.3 会話を続ける

Claude との会話は、1 回で終わりにする必要はありません。同じ画面でメッセージを続けることで、回答を深掘りしたり、修正してもらったりできます。

4.3.1 フォローアップ指示の例

もっと詳しく聞きたいとき

もう少し詳しく教えてください。
特に「〇〇」の部分が気になります。

短くしてほしいとき

この内容を3行にまとめてください。

別のアプローチを試したいとき

別の言い方で書き直してください。
もう少しカジュアルなトーンをお願いします。

追加条件を伝えたいとき

先ほどの提案書の構成に「コスト試算」のセクションも追加してください。

誤りを指摘したいとき

「〇〇」という記述は間違っています。正しくは「△△」です。
修正してもらえますか？

つまづきポイント: 「一度の指示で完璧な回答が返ってこない」と感じて Claude を諦めてしまう方がいます。実際には、会話のやり取りを重ねながら仕上げるのが Claude の正しい使い方です。最初の回答は「たたき台」と思って、どんどん修正指示を出しましょう。

4.3.2 会話の履歴は左サイドバーに残る

一度始めた会話は、左のサイドバーに一覧として残ります。後から見返したり、続きから話し始めることもできます。

新しいテーマで話すときは、左上の「**新しいチャット**」ボタンから新しい会話を始めましょう。前の会話とは別のコンテキスト（文脈）で話せます。

4.4 ファイルや画像を送る

テキストだけでなく、PDF・Word ファイル・画像なども Claude に読み込ませることができます。「この資料を要約して」「この画像に何が写っているか説明して」といった使い方が可能です。

4.4.1 ファイルの添付方法

入力欄の左下にある「**+** ボタン（またはクリップアイコン）」をクリックしてファイルを選択するか、ファイルをドラッグ&ドロップして入力欄に直接貼り付けることができます。

4.4.2 対応しているファイル形式

種類	対応形式
文書	PDF、Word (.docx)、テキスト (.txt)、CSV、HTML など
画像	JPEG、PNG、GIF、WebP
表計算	Excel (.xlsx) ※コード実行機能が必要

1 回の会話に最大 **20 ファイル**、1 ファイルあたり最大 **30MB** まで添付できます。

4.4.3 実際の使い方例

PDF の要約

添付した会議の議事録 PDF を要約してください。
決定事項と次のアクションアイテムだけを箇条書きで教えてください。

画像の説明

この画像に何が写っているか説明してください。

CSV の分析

この売上データ (CSV) を見て、前月比で伸びている商品カテゴリを教えてください。

つまづきポイント: 「ファイルが読み込めない」と感じたとき、まずファイルサイズが 30MB 以下か確認してください。また、スキャンした PDF (画像として保存された PDF) は文字を認識できない場合があります。テキストが入力されている PDF を使いましょう。

4.5 うまく伝えるコツ

Claude への指示 (プロンプト) の書き方を少し工夫するだけで、回答の質が大きく変わります。次の 3 つの基本を押さえましょう。

4.5.1 1. 具体的に書く

あいまいな指示

メールを書いて

具体的な指示

取引先 (A 社の山田部長) に対して、来週木曜日の打ち合わせの日時変更をお願いするメールを書いてください。
理由は社内会議の急な開催です。
丁寧なビジネス敬語でお願いします。

4.5.2 2. 背景（コンテキスト）を伝える

自分の立場や状況を伝えると、よりの確な回答が得られます。

私は中小企業の総務担当です。
社員 50 人の会社で、今年から在宅勤務制度を導入することになりました。
就業規則の在宅勤務規程の文案を作るのを手伝ってください。

4.5.3 3. 出力形式を指定する

「どんな形で返してほしいか」を伝えると、そのまま使いやすい形で回答が返ってきます。

競合他社 3 社の比較を表形式でまとめてください。
列は「会社名」「強み」「弱み」「価格帯」の 4 列をお願いします。

指示のポイント	悪い例	良い例
具体性	「資料を作って」	「新入社員向けの業務マニュアルの目次を作って」
背景	「メールを書いて」	「営業担当として、初回問い合わせへの返信メールを書いて」
形式	「まとめて」	「箇条書き 5 点でまとめて。各項目は 1 文以内で」

つまづきポイント: 「良い指示の書き方がわからない」という方は、まずは試してみて、気に入らなければ「もっと〇〇にして」と伝える方法が一番速いです。プロンプトを完璧にしておから送る必要はありません。

4.6 よくある疑問と注意点

4.6.1 回答が間違っていたら？

Claude は非常に高性能ですが、**事実と異なることを自信を持って述べてしまう（「ハルシネーション」と呼ばれる現象）ことがあります。**

- 重要な数字・日付・法律・医療情報などは、必ず公式ソースで確認する
- 「〇〇について確認したいのですが、出典も教えてください」と頼むと、根拠となる情報をあわせて提示してくれることがあります
- ただし、出典も誤っている場合があるため、最終確認は必ず自分で行いましょう

つまづきポイント: Claude の回答を「必ず正しい情報」として鵜呑みにするのは危険です。特にビジネスの意思決定や法的・医療的な判断が伴う場面では、Claude の回答はあくまで「参考意見」として活用し、専門家への確認も忘れずに。

4.6.2 個人情報や機密情報を入れても大丈夫？

Claude との会話はデフォルトでは Anthropic のシステムに保存されます。以下の点に気をつけましょう。

- 氏名・住所・電話番号などの個人情報 は、必要がなければ入力しない
- 社外秘の情報や未公開の事業計画 は、会社のポリシーを確認してから使う
- パスワードやクレジットカード番号 は絶対に入力しない

会社として Claude を使う場合は、管理者設定でデータの学習を無効化する等の対応もできます。詳しくは IT 部門や管理者に確認してください。

4.6.3 会話の履歴は保存される？

はい、あなたがログインしているアカウントに会話履歴が保存されます。サイドバーから過去の会話を呼び出せます。

会話を削除したい場合は、サイドバーの会話にマウスを合わせると表示される「…」メニューから「削除」を選択できます。

4.6.4 なぜ同じ質問をしても答えが変わるの？

Claude の回答には一定のランダム性があります。同じ質問でも毎回少し違う回答が返ってくることがあります。これは欠陥ではなく、より自然で多様な回答を生成するための仕組みです。

複数の案が欲しいときは「3つのパターンを提案して」と依頼するのがおすすめです。

4.7 次のステップ

Claude との最初の会話、楽しんでいただけましたか？ここで学んだ基本操作をベースに、さらに活用の幅を広げていきましょう。

Claude Web の機能をもっと使いこなしたい方:

- [Claude の料金プラン比較](#) (ページ 15) — Free プランと Pro プランの違い、アップグレードのタイミングを確認する

AI への指示をもっと上手に書きたい方:

- [プロンプトの基礎](#) (ページ 239) — AI にうまく伝えるための基本テクニックを学ぶ

ファイルを活用した業務効率化に興味がある方:

- [Projects \(プロジェクト\) 機能の使い方](#) (ページ 89) — よく使う資料や情報をまとめて管理する方法

Claude を安全に使いたい方:

- [安全な使い方・AI の限界](#) (ページ 51) — Claude を安全に使うために知っておきたいこと

Claude をもっと深く知りたい方:

- [Claude とは？— 全体像をつかもう](#) (ページ 3) — 製品ラインナップとペルソナ別のおすすめルートを確認する

第5章 Claude の料金を日本円で解説【2026年最新】

最終更新: 2026-04-07— 初版公開（消費税 10%加算対応済み）

「Claude を使ってみたいけど、実際いくらかかるの?」「ドル表示でよくわからない」「日本のクレジットカードで払えるの?」

このページでは、**2026年4月時点**の Claude の料金を日本円でわかりやすくまとめます。支払い方法、年払いの割引額、消費税の扱いまで、日本のユーザーが気になるポイントをまとめて解説します。



警告:

この記事の料金情報は **2026年4月時点** のものです。Claude の料金はドル建てのため、為替レートによって日本円換算額が変わります。最新の料金は [Claude 公式料金ページ](#) でご確認ください。

5.1 Claude の料金一覧（日本円換算）

Claude の料金はすべて**米ドル（\$）建て**です。ここでは **1ドル＝約150円** を目安に日本円換算を示します。実際の請求額は決済時の為替レートにより変わります。



注釈:

2026年4月1日から消費税が加算されます。 Anthropic は日本の消費税（10%）を全プランに加算することを発表しました。以下の表の「税込み目安」はこの消費税 10%を含んだ金額です。

5.1.1 個人向けプラン

プラン	月額（ドル）	月額目安（税抜）	月額目安（税込）
Free	\$0	無料	無料
Pro（月払い）	\$20/月	約 3,000 円	約 3,300 円
Pro（年払い）	\$17/月相当（年\$200）	約 2,550 円/月	約 2,805 円/月
Max 5x	\$100/月	約 15,000 円	約 16,500 円
Max 20x	\$200/月	約 30,000 円	約 33,000 円

1ドル＝約150円換算の目安です。 実際の請求額は決済時の為替レートによって異なります。為替レートの影響については[後述のセクション](#)（ページ 33）で詳しく説明します。

5.1.2 チーム・法人向けプラン

プラン	月額（ドル）	月額目安（税抜）
Team Standard（月払い）	\$25/席/月	約 3,750 円/席
Team Standard（年払い）	\$20/席/月	約 3,000 円/席
Team Premium（月払い）	\$125/席/月	約 18,750 円/席
Team Premium（年払い）	\$100/席/月	約 15,000 円/席
Enterprise	要問い合わせ	—

各プランの機能の違いやどのプランを選ぶべきかは、[Claude の料金プラン比較](#)（ページ 15）で詳しく解説しています。この記事では**日本円での費用と支払いの実務**に絞って説明します。

5.2 無料プランでどこまで使える？

Free プランは**完全無料**で使えます。登録するだけで、すぐに Claude と会話を始められます。

5.2.1 Free プランで使えること

- テキストでの会話・質問（日本語対応）
- メール・文章の作成・要約・翻訳
- 画像を送って内容を説明してもらう
- Web 検索（インターネットで最新情報を調べてもらう）
- 外部サービスとの連携（Slack・GitHub など 50 以上）

5.2.2 Free プランの制限

- **1 日に使える量に上限あり**: 5 時間あたり約 15～40 メッセージ程度（会話の長さや混雑状況で変わります）
- **混雑時はアクセスが遅くなる**: 平日の昼間など混雑時は制限される場合があります
- **最上位モデル（Claude Opus）は使えない**: より高性能なモデルは有料プラン限定です
- **Claude Code（AI コーディングツール）は使えない**: プログラミング支援ツールは有料プラン限定です

つまづきポイント: 「無料で使っていたら急にメッセージが送れなくなった」という声をよく聞きます。これは利用量の上限に達したためです。5 時間ほど待つとリセットされますが、頻繁に上限に当たる場合は Pro プランへの移行がおすすめです。

5.3 Pro プラン（月額約3,000 円）の費用対効果

5.3.1 月 3,000 円で何がかわるのか

Pro プランは **月額 \$20（約 3,000 円、税抜）** です。コーヒー 1 杯が 600～700 円とすると、**コーヒー約 4～5 杯分** の月額料金です。

この料金で以下が使えるようになります：

- **利用量が大幅に増える**: Free プランの約 5 倍のメッセージが送れる。1 日中使ってもほとんど上限に当たりません
- **最上位モデル「Claude Opus」が使える**: より複雑な分析、高品質な文章作成、深い推論が可能になります
- **Claude Code（AI コーディングツール）が使える**: プログラミングの自動化や、コードの作成・レビューができます
- **プロジェクト機能が使い放題**: よく使うファイルや指示をまとめて保存しておける
- **混雑時でも優先的にアクセスできる**: ピーク時でも安定して使えます

5.3.2 こんな人に Pro プランはおすすめ

- 仕事で毎日メール・報告書・資料を作成している
- 英語の文章を日本語に翻訳したり、日本語を英語に直したりすることが多い
- 調査・リサーチに使いたい
- Free プランで「もっと使いたいのにな上限に当たる」と感じている

つまづきポイント：「Pro プランにしたのに上限に当たった」という場合、主な原因は Claude Code の使いすぎです。Claude Code はチャットよりも大量のデータを処理するため、使うとすぐに消費量が増えます。開発作業がメインの方は Max プランを検討しましょう。

5.4 年払いと月払いの違い

Pro プランには **月払い** と **年払い** の 2 種類があります。

	月払い	年払い
ドル換算	\$20/月	\$200/年（約\$17/月相当）
日本円目安（税抜）	約 3,000 円/月	約 30,000 円/年（月換算約 2,500 円）
年間のお得額	—	約 6,000 円お得
支払い方法	毎月自動引き落とし	年 1 回まとめて引き落とし
解約・変更	いつでも可（当月末まで有効）	途中解約は原則返金なし



警告:

年払いは途中解約しても返金されません。「1 年間は確実に使い続ける」という確信がある場合にのみ年払いを選びましょう。迷うなら月払いからスタートして、使い続けると決めてから年払いに切り替えるのが安全です。

なお、Max プラン（\$100/月・\$200/月）には年払いオプションがありません（2026 年 4 月時点）。

5.4.1 年払いで実際にいくら節約できるか

Pro プランの場合：

- 月払い：\$20 × 12 ヶ月 = **\$240/年**（約 36,000 円/年）
- 年払い：**\$200/年**（約 30,000 円/年）
- 差額：**\$40/年（約 6,000 円）の節約**

5.5 支払い方法

5.5.1 使えるクレジットカード

Claude の有料プランは **クレジットカード払い** のみです（個人・Team プラン）。主要なカードブランドが使えます：

- Visa
- Mastercard
- American Express
- JCB（使える場合あり）

日本のクレジットカードで問題なく支払えます。 海外サービスですが、日本国内で発行されたカードでも普通に支払えます。



注釈:

法人の場合は請求書払いも可能です。Enterprise プランでは請求書払い（インボイス払い）に対応しています。また、Anthropic は日本の適格請求書発行事業者（インボイス制度対応）として登録されており、法人は仕入税額控除を受けられます（登録番号：T7700150134388）。

5.5.2 消費税の扱い

2026 年 4 月 1 日から、日本のユーザーには消費税 10%が加算されます。

- **個人利用の場合:** 料金に 10%が上乗せされます。仕入税額控除の対象外です
- **法人利用の場合:** 消費税が加算されますが、Anthropic 発行の適格請求書を使って仕入税額控除が受けられます

具体的には、Pro プランの場合：

- ・ 月払い：\$20 → 実際の請求は **\$22（消費税込み）**（日本円目安：約 3,300 円）
- ・ 年払い：\$200 → 実際の請求は **\$220（消費税込み）**（日本円目安：約 33,000 円）

5.5.3 為替レートの影響と注意点

5.5.4 ドル建て決済の仕組み

Claude の料金は米ドル (\$) で設定されており、決済もドル建てで行われます。クレジットカードに請求される日本円額は、**決済時点の為替レート**で計算されます。

つまり、1 ドル=150 円の時期と 1 ドル=160 円の時期では、同じ\$20 の請求でも日本円での負担が変わります：

為替レート	Pro プランの月額（税込\$22）
1 ドル = 140 円	約 3,080 円
1 ドル = 150 円	約 3,300 円
1 ドル = 160 円	約 3,520 円
1 ドル = 170 円	約 3,740 円

5.5.5 クレジットカードの海外事務手数料

海外サービスへの決済では、クレジットカード会社が**海外事務手数料（外貨取扱手数料）**を上乗せすることがあります。

一般的な手数料の目安：

- ・ 多くのカード：**1.6～2.2%**程度
- ・ 海外手数料無料のカード：0%（楽天カードの一部プラン、エポスカード等）

Pro プランで試算すると、手数料 2% の場合、月に約 60～70 円の上乗せになります。大きな金額ではありませんが、年間で計算すると数百円の差が出ます。

つまづきポイント：「請求明細を見たら思ったより高かった」という場合、為替レートの変動とカードの海外事務手数料が原因のことが多いです。明細には「\$22.00 → ￥3,450」のように換算後の金額が記載されます。

5.6 学生・教育機関向けの料金

5.6.1 個人の学生割引

残念ながら、個人の学生向け割引プランは現在ありません。

個人として利用する学生は、一般と同じ料金です。まずは Free プランで始めて、必要になったら Pro プランへ移行するのがおすすめです。

5.6.2 Claude for Education（大学・教育機関向け）

Anthropic は大学・高等教育機関向けに「**Claude for Education**」というプランを提供しています。これは個人が申し込むものではなく、大学や学校が機関として導入するプランです。

特徴：

- キャンパス全体（学生・教員・スタッフ）にアクセスを提供
- 機関割引での契約
- 大学側が Anthropic と直接契約を結ぶ形式

自分の大学が **Claude for Education** を導入しているかどうかは、大学の IT 部門や図書館に確認してみてください。導入済みの場合、無料または低コストで Pro 相当のアクセスが使えることがあります。



注釈:

カーネギーメロン大学など、一部の海外大学では月額 \$1 などの特別レートで学生が利用できるケースもあります。日本の大学での導入事例は増えてつつあります。

5.7 よくある質問

5.7.1 途中で解約できますか？

はい、いつでも解約できます。解約しても**当月末（または現在の契約期間の終了日）まで**は引き続き利用できます。次の請求日の前日までに解約手続きをすれば、次月は請求されません。

解約手順（Web 版の場合）：

1. claude.ai にログインする
2. 画面右上のアカウントメニューを開く
3. 「Settings（設定）」→「Billing（請求）」を開く
4. 「Cancel Plan（プランをキャンセル）」を選択する

5.7.2 返金がありますか？

原則として、**途中解約による日割り返金**は行われません。

ただし、誤って年払いに変更してしまった場合など、**誤操作（accidental purchase）**の場合はサポートに相談すると返金対応してもらえることがあります。サポートへの問い合わせは [Claude ヘルプセンター](#) から行えます。

5.7.3 プランを変更（アップグレード・ダウングレード）できますか？

はい、いつでも変更できます。

- **アップグレード**（例：Pro → Max）：即時に反映されます

- **ダウングレード**（例：Max → Pro）：次の請求サイクルの開始時から反映されます

5.7.4 日本語のサポートはありますか？

現時点では、Anthropic の公式サポートは**英語対応が中心**です。ヘルプセンター（support.anthropic.com）の記事も主に英語ですが、ブラウザの翻訳機能を使えば日本語で読めます。

チャットボットでの問い合わせは日本語でも送れますが、返答は英語になることがほとんどです。

5.7.5 法人で使う場合、インボイス（領収書）はもらえますか？

はい、Anthropic は日本の適格請求書発行事業者として登録済みです（T7700150134388）。Billing（請求）画面から領収書・請求書をダウンロードできます。法人の場合は仕入税額控除（消費税の控除）が受けられます。

5.7.6 Free プランから Pro プランにアップグレードするとデータは引き継がれますか？

はい、これまでの会話履歴やプロジェクトはそのまま引き継がれます。

5.8 次のステップ

- **Claude の料金プラン比較**（ページ 15）— Free・Pro・Max・Team・Enterprise の機能の違いを詳しく比較する
- **Claude とは？**（ページ 3）— Claude の概要と基本的な特徴を知る
- **アカウントの作成と初期設定**（ページ 9）— 無料プランで Claude を使い始める手順
- **Claude Web 版の使い方ガイド**（ページ 89）— ブラウザから Claude を活用する方法

第6章 Claude Pro の制限と対処法 — 使い切った時にやるべきこと

最終更新: 2026-04-07 — 初版公開

「Pro プランにしたのに、すぐ上限に達してしまう」「もっと使いたいのになメッセージが送れなくなった」
こういう不満、よくわかります。Pro プランに課金したのに制限が来ると、がっかりしますよね。

この記事では、Pro プランの利用制限がどのような仕組みになっているかを正確に説明した上で、**上限に当たった時に今すぐできる対処法と、制限を受けにくくする使い方のコツ**をお伝えします。

6.1 Claude Pro でできること — まず価値を確認

対処法の前に、Pro プランで何ができるかを確認しておきましょう。「そもそも Pro にする価値はあるの？」という方の参考にもなります。

6.1.1 Free プランと比べた Pro の優位点

- **利用量が約 5 倍:** Free プランと比べて、1 セッション（5 時間）あたりに送れるメッセージが大幅に増えます
- **Claude Opus が使える:** 最も高性能なモデル「Opus」にアクセスできます。複雑な分析や高品質な文章作成に最適です
- **Claude Code が使える:** ターミナルで動く AI コーディングツール「Claude Code」が利用可能になります。非エンジニアでも活用できる強力なツールです
- **優先アクセス:** 混雑している時間帯でも安定して使えます
- **プロジェクト機能（無制限）:** よく使う指示やファイルを保存して、毎回の会話で自動参照できます
- **拡張思考（Extended Thinking）:** 難しい問題に対して Claude がじっくり考えてから答える機能が使えます



注釈:

Pro プランの月額料金は \$20（年払いなら約 \$17/月相当、年 \$200 前払い）です。日常的に Claude を使うなら、Free プランの制限のストレスを考えると多くの方に合う選択肢です。

6.2 Pro プランの利用制限 — 仕組みを正確に理解する

Pro プランにも利用制限があります。ただし、Anthropic は具体的なメッセージ数を公式には公開していません。制限の仕組みを理解することが重要です。

6.2.1 2 段階の利用制限

Pro プランの制限には 2 つの軸があります。

セッション制限（5 時間ごとにリセット）

Claude は「5 時間のウィンドウ（窓）」という単位で利用量を管理しています。最初のメッセージを送った時点からカウントが始まり、5 時間後にリセットされます（毎朝 0 時などの固定時刻ではありません）。

Pro プランでは 1 セッションあたり、短めの会話なら **約 45～100 メッセージ**程度が目安ですが、この数字は会話の内容によって大きく変わります。

週次制限（7 日ごとにリセット）

セッション制限とは別に、週単位での上限もあります。週次制限は最初のメッセージから 7 日後にリセットされます。



警告:

「5 時間ごとのリセット」と「週次リセット」は別々に機能します。セッション制限がリセットされても、週次上限に達していた場合は引き続き制限がかかることがあります。

つまづきポイント: 「5 時間待てばリセットされるはずなのに、まだ送れない」という場合は、週次制限に達している可能性があります。週次制限のリセットは、最初のメッセージから 7 日後です。

6.2.2 メッセージ数に影響する要因

実際に送れるメッセージ数は、次の要因によって大きく変わります：

要因	消費量への影響
使うモデル	Opus > Sonnet > Haiku の順で消費が大きい
会話の長さ	会話が長くなるほど消費が増える
添付ファイルのサイズ	大きなファイルほど消費が増える
拡張思考（Extended Thinking）	オンにすると消費が増える
Claude Code の使用	通常のチャットより消費が大きい
ピーク時間帯	平日の混雑時間（日本時間では夜～深夜）は制限が早く来ることがある



注釈:

トークン (token) とは? Claude は「トークン」という単位でデータを処理します。日本語だと 1 文字が 1~3 トークン程度です。メッセージ数だけでなく、1 回のメッセージで扱うデータ量 (トークン数) が消費量に影響します。

6.2.3 ピーク時間の制限強化

平日の昼~夜 (日本時間で 22 時~翌 4 時頃) は、世界的なアクセスが集中するピーク時間にあたります。この時間帯は同じ 5 時間でも制限の消費ペースが速くなることがあります。急いでいない作業は空いている時間帯 (日本の早朝や週末) に行うと効果的です。

6.3 「上限に達しました」が出た時の対処法

上限メッセージが表示された時に試せる方法を、即効性の高い順に紹介します。

6.3.1 1. Haiku モデルに切り替える

Claude Web の画面右上にあるモデル選択から、**Claude Haiku** に切り替えると、消費量を大幅に抑えながら会話を続けられます。

Haiku は Opus や Sonnet に比べて応答速度が速く、シンプルな質問や軽い作業には十分な性能があります。「今すぐ回答が欲しい」という場合にまず試すべき方法です。

6.3.2 2. 新しい会話を始める

同じ会話を長く続けると、会話の履歴全体がトークンとして消費されます。**新しいチャット画面から会話を始め直す**と、消費量をリセットできます。

「前の会話の文脈を引き継ぎたい」場合は、重要な情報だけを要約してコピーし、新しい会話の最初に貼り付けると効率的です。

6.3.3 3. 5 時間待つ

急ぎでない場合は、セッション制限がリセットされるまで待つのがシンプルな方法です。最初のメッセージから 5 時間後にリセットされます。

Claude の画面にはリセットまでの残り時間が表示されることがあるので確認してみましょう。

6.3.4 4. 会話を要約してから続ける

上限が近づいてきたと感じたら、Claude に「これまでの会話の要点を箇条書きでまとめて」と依頼しましょう。その要約を新しい会話に貼り付けることで、長い会話履歴を引き継がずに会話を継続できます。

6.3.5 5. Extra Usage（追加利用）を購入する

Max プランへのアップグレードが難しい場合、「**Extra Usage**」という追加利用オプションがあります。上限に達した後も追加料金を支払って利用を続けられる機能です。Anthropic の設定画面から有効にできます。



警告:

Extra Usage は API と同じ単価（トークン消費量ベース）で課金されます。頻繁に必要ななら Max プランへの切り替えのほうがコストを抑えられることがあります。

6.4 利用量を節約する 5 つのコツ

制限に当たる頻度を減らすための、日常的な使い方のコツです。

6.4.1 コツ 1: Sonnet を標準として使い、Opus は必要な時だけ

Claude Opus は最も高性能なモデルですが、消費量も最大です。日常的な質問、文章の添削、調べ物などは **Sonnet** で十分対応できます。

Opus を使うべき場面：

- 複雑な分析や高度な推論が必要な時
- 長い文書の精密なレビュー
- 専門的な判断が求められる場面

6.4.2 コツ 2: 会話を適切に区切る

「ちょっと聞きたいことが増えた」と同じ会話をずっと続けると、会話の蓄積がどんどん重くなります。**テーマが変わったら新しい会話を始める**習慣をつけましょう。

目安として、会話が 20〜30 往復を超えてきたら一度切り上げて新しい会話にするのがおすすめです。

6.4.3 コツ 3: 添付ファイルのサイズを絞る

PDF や文書を添付する時、ファイル全体を渡すよりも**関係する部分だけを抜き出してテキストで貼り付け**ると消費量を抑えられます。

例えば、50 ページの PDF のうち 3 ページだけが必要なら、そのページのテキストをコピーして貼り付けるほうが効率的です。

6.4.4 コツ 4: Extended Thinking は本当に必要な時だけオンに

「拡張思考 (Extended Thinking)」は Claude が問題をじっくり考える機能で、消費量が通常の 2〜3 倍になることがあります。難しい数学の問題や複雑な戦略的判断など、精度が重要な場面に絞って使いましょう。

6.4.5 コツ 5: 1 回のメッセージで複数の質問をまとめる

「まずこれを聞いて、その答えに応じて次を聞いて…」と 1 問ずつ聞くより、**関連する質問をまとめて 1 回のメッセージで送る**と会話の往復回数を減らせます。

良い例：

以下の 3 点について教えてください。

1. ○○の概要と背景
2. ○○のメリット・デメリット
3. ○○を導入する場合の具体的なステップ

6.5 Claude Code と利用枠の関係

Claude Code（ターミナルで動く AI コーディングツール）を使っている場合、特に注意が必要です。

6.5.1 Web 版と利用枠を共有している

Claude Code は Claude Web（ブラウザ版）と**同じ利用枠を消費します**。Claude Code でたくさん作業した後に Web 版で会話しようとする、すでに上限に近い状態になっていることがあります。

6.5.2 Claude Code はトークン消費が大きい

通常のチャットと比べて、Claude Code のコーディング作業は大量のトークンを消費します。理由は：

- ファイルの内容を読み込む（ファイル 1 つで数千～数万トークン）
- コードの実行結果を受け取る
- 複数のファイルを同時に扱う

目安として、Claude Code で 1 時間作業すると、Web 版チャットで 10～30 往復した場合と同程度の消費になることがあります（作業内容による）。

6.5.3 Pro プランで Claude Code はどのくらい使える？

Pro プランで Claude Code を使う場合の現実的な目安：

- **軽い作業（小さなファイルの修正、1～2 ファイルの作成）**：1 日数回は問題なく使えることが多い
- **中規模の作業（機能の追加、複数ファイルの修正）**：セッションの消費が早くなる。1 日 1～2 セッションが目安
- **大規模な作業（リファクタリング、大量ファイルの処理）**：すぐに上限に達しやすい。Max プランを検討する

つまづきポイント：「Pro プランで Claude Code を使い始めたら、あっという間に制限が来た」というのはよくある声です。Claude Code は想像以上にトークンを消費します。コーディングがメインの利用なら、最初から Max プランを検討するのが結果的にストレスが少なくなります。

6.6 Max プランに切り替えるべきタイミング

「Pro では足りないかもしれない」と感じたら、以下の基準を参考にしてください。

6.6.1 Max プランを検討すべきサイン

利用量の観点から：

- ・ 週に 3 回以上、セッション制限に達している
- ・ 毎日 Claude Code を 2～3 時間以上使っている
- ・ ピーク時間帯に仕事で使う必要があり、頻繁に制限に当たる

作業内容の観点から：

- ・ 大規模なコードベースのリファクタリングを Claude Code で行いたい
- ・ 50 ページ以上の文書を毎日分析している
- ・ Deep Research（詳細な調査レポート）を頻繁に使っている

6.6.2 Max プランの選び方

Max プランには 2 段階あります：

	Max 5x	Max 20x
月額	\$100	\$200
Pro 比の利用量	5 倍	20 倍
こんな人向け	Pro の上限に週 3 回以上当たる人	Claude Code をほぼ一日中使う人

判断のコツ： まず Max 5x を試して、それでも上限に当たるなら Max 20x を検討する、というステップアップが無駄が少ないです。



注釈：

Max プランでも上限はあります。ただし Pro の 5～20 倍の量なので、日常的な業務であれば Max 5x で上限に当たることはほとんどありません。

6.7 Free vs Pro vs Max 利用量比較

3 つのプランの利用量を整理します。

	Free	Pro	Max 5x	Max 20x
月額	無料	\$20	\$100	\$200
セッション制限	あり（低め）	あり（Free の 5 倍）	あり（Pro の 5 倍）	あり（Pro の 20 倍）

次のページに続く

表 3 – 前のページからの続き

	Free	Pro	Max 5x	Max 20x
週次制限	あり	あり	あり	あり
Claude Opus	利用不可	利用可	利用可	利用可
Claude Code	利用不可	利用可	利用可	利用可
優先アクセス	なし	あり	あり（最優先）	あり（最優先）
Extra Usage	購入不可	購入可	購入可	購入可

料金に関する注意: この表の料金情報は 2026 年 4 月時点のものです。最新の情報は [Anthropic 公式の料金ページ](#) で確認してください。

6.8 よくある質問

6.8.1 上限のリセットはいつ？

セッション制限は**最初のメッセージから 5 時間後**にリセットされます（毎朝 9 時などの固定時刻ではありません）。また、週次制限は**最初のメッセージから 7 日後**にリセットされます。

6.8.2 Opus と Sonnet で消費量は違う？

Anthropic は具体的な数字を公開していませんが、一般的に **Opus は Sonnet の 3～5 倍程度のトークンを消費する**とされています。同じメッセージ数でも、Opus を使い続けると Sonnet より早く上限に達します。

日常的なタスクには Sonnet、精密さが重要な場面には Opus というメリハリのある使い方がおすすめです。

6.8.3 API と利用枠は別？

はい、まったく別です。

- **Claude Web / Desktop / Code の利用枠:** Pro ・ Max プランの月額に含まれる
- **API (Anthropic Console) の利用枠:** プランとは独立した別払い（使った分だけ課金）

API は開発者がプログラムから Claude を呼び出す仕組みで、Pro プランに加入しているからといって API が自動的に使えるわけではありません。API を使いたい場合は [Anthropic Console](#) で別途アカウントを作成する必要があります。

6.8.4 平日の夜に制限が早く来るのはなぜ？

Claude のサーバーは世界中からアクセスされています。平日の午前～夕方（米国太平洋時間）は最もアクセスが集中するため、この時間帯（日本時間で 22 時～翌 4 時頃）は制限の消費ペースが速くなることがあります。急ぎでない作業は日本の午前中（平日 9～17 時）に行うと比較的空いていることが多いです。

6.8.5 Claude Code を使うと Web の利用枠が減る？

はい、Claude Code と Claude Web（ブラウザ版）は同じ利用枠を共有しています。Claude Code で大きなコーディング作業をした後に Web 版でチャットしようとする、すでに上限に近い状態になっていることがあります。

6.8.6 Max プランにして合わなかったら戻れる？

はい、いつでもプランを変更できます。ダウングレードは次の請求サイクルから適用されます。「まず 1 ヶ月 Max 5x を試して、余裕があれば Pro に戻す」という使い方も可能です。

6.9 次のステップ

- [Claude の料金プラン比較](#) (ページ 15) — Free・Pro・Max・Team・Enterprise を全プラン比較。自分に合ったプランの選び方
- [Claude とは？— 全体像をつかもう](#) (ページ 3) — Claude の製品ファミリーと使い分けを理解する
- [プロンプトの基礎](#) (ページ 239) — 少ないメッセージ数で欲しい回答を引き出すコツ。利用量節約にも直結する

第7章 Claude のモデル比較・選び方 — Opus・Sonnet・Haiku どれを使う？

最終更新: 2026-04-09

Pro プランを契約したはいいいけれど、「Opus・Sonnet・Haiku って何？どれを使えばいいの？」と迷っていませんか？

この記事はそんな疑問に答えます。**どのプランを選ぶかは別の記事（料金プラン比較（ページ 15））**で解説しています。この記事では「Pro プラン以上を持っている方が、日々の作業に合わせてどのモデルを選ぶか」を詳しく説明します。



注釈:

この記事の前提: Pro プラン以上（Pro・Max・Team・Enterprise）で使える全モデルの比較です。Free プランの方は Sonnet と Haiku のみ使えます。まずプランを選びたい方は [Claude の料金プラン比較](#)（ページ 15）をご覧ください。

7.1 3つのモデル早わかり表

Claude には現在 3 つのモデルがあります。すべて「Claude（クロード）」という同じ AI ですが、**性能・速度・消費量のバランスが異なります**。料理で例えると、同じシェフが「じっくり仕上げる本格料理」「手早く作る定食」「超スピードの軽食」をそれぞれ担当するイメージです。

モデル	読み方	一言説明	性能	応答速度	利用量の消費	向いているタスク
Opus	オーパス	最高性能・深く考える	最高	やや遅い	多い	複雑な分析、長文レビュー、高度な推論
Sonnet	ソネット	バランス型・日常の万能選手	高	速い	標準	日常業務全般、文書作成、調査
Haiku	ハイク	最速・軽量	標準	最速	少ない	翻訳、簡単な質問、大量処理

つまづきポイント: 「一番いい Opus をずっと使えばいいじゃないか」と思うかもしれません。ただし Opus は消費量が多いため、1 日に使えるメッセージ数が減ります。目的に合わせて使い分けることで、利用上限を気にせず快適に使えます。

7.2 Opus（オーパス） — じっくり考える最高性能モデル

Claude Opus 4.6 は、現在提供されている最も高性能なモデルです。複雑な問題に対して、多角的な視点で深く考えてから答えを出します。

7.2.1 Opus が得意なこと

- **複雑な分析と推論** — 複数の条件が絡み合う問題の整理
- **高品質な文章作成** — 論理的な構成と洗練された表現を求められる文書
- **長文の精密なレビュー** — 細かいニュアンスまで読み取る必要がある場面
- **専門的な判断** — 曖昧さを含む状況での慎重な見解の提示

7.2.2 Opus を使うべき具体的な場面

契約書・法的文書のリスク確認 「この 30 ページの業務委託契約書を読んで、リスクになる条項をすべて洗い出して」のような依頼。文脈を慎重に読み取る必要がある場合に Opus の精度が光ります。

複雑なビジネス戦略の検討 「競合他社 A・B・C の戦略を比較した上で、自社が参入すべき市場セグメントを提案して」のような多段階の推論が必要なタスク。

高品質な文章の作成 役員向けの報告書、重要な顧客向け提案書など、品質を妥協できない文書の作成。

7.2.3 Opus を使う時の注意点

- **応答に時間がかかる**: Sonnet と比べると、回答が出るまでやや待ちます（複雑な質問なら数十秒のこともあります）
- **消費量が多い**: 同じ内容のやり取りでも、Opus は Sonnet より多くの消費量がかかります。利用制限に達するのが早くなります
- **シンプルなタスクには不要**: 「この文章を要約して」「英語に翻訳して」程度の作業では Sonnet や Haiku と大差ありません



警告:

Opus は Pro プラン以上でのみ使えます。Free プランでは利用できません。また、消費量が多いため、1 日中 Opus を使い続けると利用制限に早く当たります。大事な作業に絞って使うのがコツです。

7.3 Sonnet（ソネット） — 日常使いのバランス型

Claude Sonnet 4.6 は、性能と速度のバランスが最も優れたモデルです。「迷ったらまず Sonnet」が Claude を上手く使うための基本方針です。

7.3.1 Sonnet が得意なこと

- **日常的な文書作成** — メール、報告書、議事録など
- **情報の整理・要約** — 長い文章や複数の資料をまとめる
- **アイデア出し・ブレインストーミング** — 幅広い発想を素早く展開
- **調査・リサーチの補助** — 特定のテーマについて情報を整理する

- コードの簡単な修正・説明 — 基本的なプログラミング補助

7.3.2 Sonnet を使うべき具体的な場面

会議の議事録作成 「この会議メモから議事録を作って。決定事項と次のアクション項目を明確にして」のような依頼。毎日やる作業なら Sonnet で十分です。

メールの下書き作成 「先方への提案お断りメールを、丁寧かつ関係を壊さないトーンで書いて」のような日常的なビジネス文書。

資料の要約と整理 「この 10 ページのレポートを、経営層向けに 1 ページのエグゼクティブサマリーにまとめて」という作業。

7.3.3 「迷ったらまず Sonnet」の理由

多くのベンチマークで Sonnet は Opus とほぼ同等のスコアを出しており、日常業務では差を感じにくいレベルです。一方で、PhD レベルの高度な推論など一部の難しいタスクでは Opus が明確に上回ります。消費量は Sonnet のほうが少ないため、まず Sonnet を使い、「もっと精度が欲しい」と感じた時だけ Opus に切り替えるのが効率的です。

7.4 Haiku（ハイク） — 素早く答える軽量モデル

Claude Haiku 4.5 は、3つのモデルの中で最も応答が速く、消費量が少ないモデルです。名前の「俳句」のように、短く素早く答えてくれます。

7.4.1 Haiku が得意なこと

- 翻訳・言語変換 — 文章を別の言語や表現に変える
- シンプルな質問への回答 — 調べ物、用語の意味確認など
- テキストの簡単な加工 — 箇条書き化、フォーマット変換など
- 大量の短いタスク処理 — 多数の似た作業を次々とこなす
- 利用制限に近づいた時の代替 — Sonnet や Opus の利用上限が近い時の繋ぎ

7.4.2 Haiku を使うべき具体的な場面

英文メールの翻訳 「この英語のメールを日本語に訳して」のようなシンプルな翻訳は、Haiku で十分かつ素早く完了します。

単語・用語の確認 「"due diligence"ってどういう意味？ビジネス文脈で教えて」のような一問一答。

文章のフォーマット変換 「この箇条書きリストを表にして」「この文章を敬体（ですます調）から常体（だ・である調）に変えて」のような機械的な変換作業。

つまづきポイント: 「Haiku は性能が低いから使わなくていい」と思っていないですか？単純な作業なら Haiku は Sonnet とほぼ同等の結果を出します。しかも消費量が少ないので、利用制限に近づいた時に Haiku に切り替えると作業を続けられます。

7.5 用途別おすすめ早見表

「どれを使えばいいか迷ったら」のクイックガイドです。

やりたいこと	おすすめモデル	理由
契約書・法的文書の確認	Opus	細かいニュアンスまで読み取る精度が必要
重要な提案書・企画書の作成	Opus	高品質な論理構成と表現を求める場合
複雑なデータ分析・解釈	Opus	多角的な視点と深い推論が必要
日常的なメール・文書作成	Sonnet	速度と品質のバランスが最適
会議の議事録まとめ	Sonnet	構造化と要約が得意で十分な精度
市場調査・情報収集補助	Sonnet	広範な知識と整理力を活かせる
プレゼン資料の構成案	Sonnet	アイデア展開と構造化に向いている
英語・他言語の翻訳	Haiku	翻訳は Haiku で十分な精度が出る
単語・用語の意味確認	Haiku	素早い回答が得られる
大量の短い文章を変換する	Haiku	消費量が少なく大量処理に向く
制限が近づいた時の作業継続	Haiku	少ない消費量で作業を続けられる

7.6 モデルの切り替え方

7.6.1 claude.ai でのモデル選択

claude.ai のチャット画面では、送信ボタンの横にあるモデル名をクリックするだけでモデルを切り替えられます。なお、モデルを変更すると**新しいチャットが始まります**（それまでの会話は別のチャットとして残ります）。

モデル選択のドロップダウンには現在使えるモデルの一覧が表示されます。各モデル名の横に簡単な説明が表示されるので、参考にしながら選べます。



注釈:

「自動」設定について: モデル選択に「自動 (Auto)」というオプションがある場合、Claude がタスクの複雑さに応じて自動的に適切なモデルを選びます。迷ったら「自動」に設定しておくのも一つの方法です。ただし、消費量の予測がしにくくなるため、利用上限を気にする方は手動でモデルを指定することをおすすめします。

7.6.2 Free プランでの制限

Free プランでは **Haiku と Sonnet のみ**が使えます。Opus を使いたい場合は Pro プラン以上へのアップグレードが必要です。

7.6.3 モデル選択のコツ

1. **最初は Sonnet から始める** — まず Sonnet で試して、もっと精度が必要と感じた時に Opus を試す
2. **繰り返し作業は Haiku で節約** — 同じような翻訳や変換を大量にこなす時は Haiku で OK
3. **重要な作業は Opus で仕上げる** — 最終確認や高品質が求められる場面で Opus を使う

7.7 番外編: Mythos — セキュリティ研究向けモデル

2026 年 4 月、Anthropic は「**Claude Mythos Preview**」という新しいモデルを発表しました。このモデルは通常の Opus・Sonnet・Haiku とは異なる、特殊な位置づけのモデルです。

7.7.1 Claude Mythos とは

Mythos は、コンピューターセキュリティ分野、特に**脆弱性（セキュリティ上の欠陥）の発見と分析**において突出した能力を持つモデルです。Anthropic の発表によれば、Mythos はわずか数週間で主要なオペレーティングシステムやブラウザに存在する多数のゼロデイ脆弱性（未発見の欠陥）を自律的に発見したとのこと。

7.7.2 Project Glasswing

Mythos の提供は「**Project Glasswing**」という取り組みを通じて行われています。Anthropic を含む 12 社 — Amazon Web Services (AWS)、Apple、Broadcom、Cisco、CrowdStrike、Google、JPMorganChase、Linux Foundation、Microsoft、NVIDIA、Palo Alto Networks が参加し、防衛的なセキュリティ用途（自社システムの脆弱性を先に発見して修正する目的）での活用が認められています。

詳細は [Anthropic の Project Glasswing ページ](#) をご参照ください。

7.7.3 一般ユーザーは使えない

Mythos は現在、一般向けには提供されていません。Anthropic は自己申込みによるアクセスを提供しておらず、招待制の限定プレビューとして運用されています。

将来的には安全性の確保が進んだ段階でより広く提供されることを目指しているとのことですが、2026 年 4 月時点では一般ユーザーが使える状況ではありません。



注釈:

一般ユーザーが claude.ai やアプリで使えるモデルは、引き続き Opus・Sonnet・Haiku の 3 つです。Mythos の存在は「こんなモデルもある」と知っておく程度で OK です。

7.8 よくある質問

7.8.1 Opus を使うと利用制限が早く来る？

はい、Opus は Sonnet に比べて消費量が多いため、同じ利用上限の中でより早く制限に達します。具体的な倍率は公表されていませんが、体感としては Sonnet よりかなり早く制限に当たります。

対策は 3 つあります：

1. **Opus は重要な作業に絞り**、日常業務は Sonnet や Haiku を使う
2. **利用制限に近づいたら Haiku に切り替えて**作業を続ける
3. 頻繁に制限に当たるなら **Max プランへのアップグレードを検討する**

7.8.2 「自動」設定のままでいい？

「自動」設定は Claude が最適なモデルを選んでくれるので、迷う手間がなくなります。ただし、自動設定では消費量の予測がしにくく、気づかずに Opus が多用されて制限に早く当たるケースもあります。

おすすめの使い方: 最初は Sonnet を手動で選んでおき、「もっと深く考えてほしい」と感じた時だけ Opus に切り替えるのが、消費量をコントロールしやすくなります。

7.8.3 Sonnet と Haiku はどのくらい違う？

日常的な翻訳・要約・文章変換などのシンプルな作業では、**ほぼ差がわからない**ことがほとんどです。複雑な推論や高品質な文章作成になると Sonnet のほうが明らかに優れた結果を出します。

迷ったらまず Sonnet で試して、「もっと速く処理したい」「利用量を節約したい」という場合に Haiku を使うのが無難です。

7.8.4 モデルのバージョン（4.6 とか 4.5 とか）は気にする？

一般ユーザーが `claude.ai` で使う場合は、**バージョン番号を気にする必要はありません**。常に最新バージョンが自動的に使われます。「Opus」「Sonnet」「Haiku」の 3 種類を使い分けることだけ覚えておけば大丈夫です。

バージョン番号（4.6、4.5 など）は、開発者が API でプログラムから Claude を呼び出す際に、特定のバージョンを指定するために使われるものです。

7.8.5 スマホアプリでもモデルを切り替えられる？

はい、Claude のスマホアプリ（iOS / Android）でもモデルの切り替えができます。チャット画面の上部にあるモデル名をタップすると選択画面が開きます。

7.8.6 Haiku は Free プランでも使える？

はい、Haiku は Free プランでも使えます。Sonnet と Haiku は Free プランで利用可能です。Opus は Pro プラン以上が必要です。

7.9 次のステップ

- [Claude の料金プラン比較](#) (ページ 15) — どのプランを契約するか迷っている方はこちら
- [Claude Pro の制限と対処法](#) (ページ 37) — 利用制限に頻繁に当たる場合の対処法
- [Claude Web 版の使い方ガイド](#) (ページ 89) — `claude.ai` の基本操作を学ぶ
- [プロンプトの基礎](#) (ページ 239) — AI にうまく伝える指示の出し方

第8章 安全な使い方・AIの限界

「Claudeは何でもできる万能AI?」「会話の内容って外に漏れないの?」「AIが教えてくれたことをそのまま信じていいの?」

Claudeを使い始めると、こんな疑問が出てくることがあります。このページでは、**Claude（およびAI全般）の限界とリスクを正しく理解して、安全に賢く使うための知識**をお伝えします。

Claudeはとても優秀なAIアシスタントですが、万能ではありません。特性と限界を理解することが、より良い使い方への第一歩です。

8.1 AIは万能ではない

Claudeは驚くほど多くのことができますが、**苦手なことや、できないことも明確にあります**。まずその境界線を押さえておきましょう。

8.1.1 Claudeが得意なこと

- **文章の作成・編集** — メール、報告書、プレゼン資料のドラフト作成
- **情報の整理・要約** — 長い文書を短くまとめる、ポイントを抽出する
- **アイデア出し** — ブレインストーミング、企画の壁打ち
- **説明・解説** — 難しい概念をわかりやすく説明する
- **翻訳** — 日英をはじめとした多言語翻訳
- **コード作成** — プログラムのコードを書いたり、エラーを調べたりする

8.1.2 Claudeが苦手なこと・できないこと

- **リアルタイムの最新情報の取得** — Claudeの知識には学習データの期限（カットオフ）があります。「今日の株価」「最新ニュース」などはわかりません
- **Web検索の正確性** — ClaudeにはWeb検索機能がありますが、検索結果が常に正確・最新とは限りません。重要な情報は自分でも公式サイト等で確認しましょう
- **画像・動画・音声の生成** — テキストと画像の読み込みはできますが、新しい画像や動画を作ることはできません
- **感情を本当に理解すること** — 言葉としては共感を示せますが、人間のような感情体験はありません

つまづきポイント: Claude の Web 検索機能で調べた情報をそのまま信じていませんか？ Web 検索の結果は必ずしも正確・最新とは限りません。また、検索を使わずに学習済みデータだけで回答する場合があります。最新情報や重要な情報は、必ず公式サイトや信頼できる情報源で自分でも確認しましょう。

8.2 ハルシネーション——もっともらしい嘘に注意

AI の最も重要なリスクのひとつが「ハルシネーション」です。

ハルシネーション (Hallucination) とは、AI が事実ではない情報を、あたかも正確であるかのように自信満々に答えてしまう現象です。「幻覚」と訳されることもあります。

8.2.1 具体的な例

たとえばこんなことが起こります。

- 存在しない本や論文を「〇〇という著書があります」と引用する
- 架空の人物の経歴を詳細に語る
- 数字の計算を間違えるが、正確そうに答える
- 法律や制度について古い情報や誤った情報を答える

Claude 自身もこのリスクを認識しており、「わからないことは正直にわからないと言う」よう設計されています。ただし、完全にハルシネーションをなくすことは技術的にまだ難しく、**どんな AI にも存在するリスク**です。

8.2.2 ハルシネーションへの対策

1. **重要な事実は必ず裏を取る** — 数字、人名、法律、医薬品名など、重要な情報は公式サイトや文献で確認する
2. **「出典を教えてください」と聞く** — 情報源を提示してもらうことで、確認しやすくなる。ただし、示された URL や書名が実在するかも確認が必要
3. **不確かな場合は聞き直す** — 「この情報は確かですか？」「自信はどのくらいありますか？」と追加で確認する
4. **専門性の高い分野は専門家に相談** — 法律、医療、金融の重要な判断は専門家の確認を必ず

つまづきポイント: 「Claude が教えてくれたから正しいはず」と思い込んでしまうのが一番危険です。Claude は自信を持って間違えることがあります。「それっぽい文章」でも事実とは限らないと覚えておきましょう。

8.3 個人情報・機密情報の扱い

「Claude との会話って、外部に漏れないの?」という疑問は当然です。ここで仕組みを正しく理解しておきましょう。

8.3.1 会話データはどう扱われる？

Anthropic は 2025 年 9 月にプライバシーポリシーを更新し、**プランによってデータの取り扱いが異なります**。

プラン	データの取り扱い
Free / Pro / Max	デフォルトでは会話データを AI 改善のために使用する場合があります。設定でオプトアウト可能
Claude for Work (Team / Enterprise)	会話データはモデル学習に使用されない
API 経由の利用	会話データはモデル学習に使用されない

Free・Pro・Max プランを使っている方は、**設定 → データとプライバシー** からデータ共有をオフにできます。

8.3.2 入力すべきでない情報

以下のような情報は、Claude の会話に入力しないことを強くおすすめします。

個人情報:

- マイナンバー、パスポート番号、運転免許証番号
- クレジットカード番号、銀行口座番号
- 本人や他者の住所、電話番号、生年月日

機密情報:

- 社外秘の業務資料、顧客情報
- 未公開の経営情報、財務データ
- 取引先の機密情報（守秘義務のあるもの）

パスワード・認証情報:

- 各種サービスのパスワード
- API キー、システムのアクセス情報

つまづきポイント: 「このメールの内容を要約して」と貼り付けたときに、顧客の個人情報が含まれていませんか？ コピペするときは、不必要な個人情報が混入していないかを確認する習慣をつけましょう。

8.4 著作権・知的財産

Claude を使って文章や画像、コードを作成するとき、**著作権に関する注意が必要な**ケースがあります。

8.4.1 Claude が生成したコンテンツの著作権

現時点（2026 年）の日本の法律では、AI が生成したコンテンツの著作権については法整備が進んでいる段階です。一般的に、AI が生成した文章そのものには著作権は発生しないとされることが多いですが、法律の解釈は変わる可能性があります。

重要なのは、「Claude に作ってもらったから自分のオリジナル」と言い切れないケースがあることです。特に商業利用やコンペへの応募等では、生成コンテンツの利用規約を確認しておきましょう。

8.4.2 他者の著作物を入力するときの注意

- **書籍・記事の全文コピペは避ける** — 一部の引用や要約のために使うのは一般的ですが、全文を無断で大量に入力することは著作権侵害になる可能性があります
- **社内資料の外部サービスへの入力** — 会社の規定で外部 AI サービスへの情報提供が制限されている場合があります。就業規則や情報セキュリティポリシーを確認しましょう
- **生成コンテンツの商用利用** — 広告や製品説明などへの利用は、事前に確認が必要な場合があります

つまづきポイント: 「Claude に作ってもらった文章をそのまま提出したら問題ない?」これは用途によります。社内文書であれば多くの場合問題ありませんが、学術論文、コンペ、著作権が関わる場面では利用規約や規定を確認してください。

8.5 バイアスと公平性——AI の回答には偏りがある

AI は学習データをもとに回答を生成します。そのため、**学習データに含まれていた偏り（バイアス）**が回答に反映されることがあります。

8.5.1 バイアスが生じやすい場面

- **文化・地域による違い** — 日本と海外では常識や慣習が異なります。Claude の回答は特定の文化や地域の視点に偏っている可能性があります
- **時代による違い** — 学習データには古い情報も含まれており、古い価値観が反映されることがあります
- **特定の立場への偏り** — 政治、宗教、倫理的に議論のあるテーマでは、特定の立場の意見が強く出ることがあります

8.5.2 バイアスへの対処法

- **複数の視点を求める** — 「賛成意見と反対意見を両方出して」「別の見方はある?」と聞く
- **重要な判断には複数の情報源を使う** — 採用、投資、重要な意思決定に AI だけを使わない
- **回答を鵜呑みにしない** — 「Claude がそう言ったから」ではなく、自分でも考える習慣を持つ

8.6 AI に任せてはいけないこと

Claude は多くのことを手伝えますが、**最終的な責任が発生する判断は必ず人間が行う必要があります**。

8.6.1 人間の確認・判断が必須な領域

法的判断: Claude に法律の解釈を聞くことはできますが、「法的にこれで大丈夫か」という最終判断は弁護士に相談してください。契約書の作成や法的な効力が必要な文書は、専門家のレビューが必要です。

医療アドバイス: 症状の解説や一般的な情報は提供できますが、診断や治療法の選択は医師に相談してください。「Claude が大丈夫と言っていたから」という理由で受診を遅らせることは危険です。

財務・投資判断: 投資戦略の考え方を相談することはできますが、具体的な投資の判断は専門のファイナンシャルプランナーや証券会社に相談してください。資金の運用は自己責任が伴います。

重大な人事・組織判断: 採用・解雇・評価などの判断に AI を参考にするには問題ありませんが、最終的な判断は人間が責任を持って行ってください。個人情報が含まれる場合は特に注意が必要です。

つまづきポイント: 「Claude がすすめてくれたから」という理由だけで重要な判断をしてしまう人がいます。AI はあくまで「参考意見を出すツール」であり、最終的な責任を取るのは人間です。「これで大丈夫かな?」と思ったら、Claude ではなく専門家に確認する習慣をつけましょう。

8.7 Claude の安全設計について

ここまでリスクや注意点を伝えしてきましたが、Claude は「安全であること」を最も重視して設計されている AI のひとつです。

Anthropic は「**Constitutional AI (憲法的 AI)**」と呼ばれる手法で Claude を訓練しています。これは、Claude が守るべき価値観や行動原則を明文化した「憲法」をもとに、AI が有害な回答をしないよう訓練する仕組みです。Claude の「憲法」は 2026 年 1 月に公開されており、誰でも内容を確認できます。

Claude が守っている主な原則:

- **正直であること** — わからないことをわからないと伝え、嘘をつかない
- **有害なコンテンツを避けること** — 危険な情報、差別的な内容、違法な情報は提供しない
- **人間の監督を支持すること** — AI が人間の判断を超えることをしない

これらの設計があっても完璧ではありませんが、Claude が「安全に使える AI」であるよう継続的に改善されていることは理解しておくくと安心です。

8.8 安全に使うためのチェックリスト

最後に、日々の Claude 利用で意識したいポイントをまとめます。

情報の扱い:

- ☐ 重要な数字・事実（法律、医療情報、統計など）は必ず公式情報源で裏を取る
- ☐ マイナンバー、クレジットカード番号、パスワードなどは入力しない
- ☐ 顧客情報、社外秘の情報は入力しない

- ☐ 「情報の出典を教えて」と確認する習慣をつける

判断・意思決定:

- ☐ 法律、医療、財務の重要な判断は専門家に相談する
- ☐ Claude の回答を「参考意見」として使い、最終判断は自分で行う
- ☐ 複数の視点を求めて、一方的な意見に偏らないようにする

プライバシー設定:

- ☐ Free / Pro / Max プランなら、[データとプライバシー設定](#)を確認する
- ☐ 会社でのご利用は、[情報セキュリティポリシー](#)を確認する

著作権・利用規約:

- ☐ 商業利用や公式なコンテンツへの利用は、目的に合った利用規約を確認する
- ☐ 他者の著作物を大量に入力することは避ける

このページでお伝えしたいことは、「Claude を使うのをやめるべきだ」ではありません。正しく特性を理解したうえで使うことで、Claude はとても心強いアシスタントになります。「賢いツールとして活用する、ただし最終的な確認と判断は自分で」——この姿勢が安全で効果的な AI 活用の基本です。

8.9 関連リンク

- [Claude とは? — 全体像をつかもう](#) (ページ 3) — Claude の製品と機能の全体像
- [Claude の料金プラン比較](#) (ページ 15) — Free から Enterprise まで、プランごとの違い
- [プロンプトの基礎](#) (ページ 239) — Claude により正確に伝えるための方法

第9章 Claude と ChatGPT の違いを徹底比較【2026年最新】

最終更新: 2026-04-07

「ChatGPT と Claude、どっちを使えばいいの?」「両方使ってみたいけど、違いがよくわからない」

このページでは、**2026年4月時点**の情報をもとに、Claude と ChatGPT を料金・性能・得意分野の3つの軸で徹底的に比較します。どちらが自分に合っているかを判断できるよう、公平に解説します。



警告:

この記事の情報は **2026年4月時点** のものです。AI サービスの料金・機能は頻繁に変更されます。最新情報は [Claude 公式サイト](#) と [ChatGPT 公式サイト](#) でご確認ください。

9.1 結論：3秒でわかる比較表

急いでいる方のために、まず結論をまとめます。

用途	おすすめ	理由
プログラミング・コーディング	Claude	Claude Code (ターミナル統合) による自律的なコーディングが強み
長文の文章作成・編集	Claude	100万トークンの文脈処理、自然な文体
画像を生成したい	ChatGPT	Claude には画像生成機能がない (ChatGPT は DALL-E 内蔵)
音声会話をしたい	ChatGPT	高品質な音声モード搭載 (Claude もベータ提供中)
日常的なチャット・雑談	どちらでも	無料プランでどちらも試せる
ビジネス文書の作成	Claude	長い文章の理解と自然な日本語に強み
データ分析・推論	Claude 寄り	長い文脈での論理的・一貫性に強み
多様なツール連携	ChatGPT	GPTs (カスタム AI) やプラグインエコシステムが充実

つまづきポイント: 「どちらが優れているか」という問いに正解はありません。Claude はコーディングと文章作成の深度で勝り、ChatGPT は対応できる機能の幅 (画像生成・音声・ツール連携) で勝ります。用途によって使い分けるのが最も賢い使い方です。

9.2 基本情報の比較

まず、両サービスの基本的な情報を整理します。

項目	Claude	ChatGPT
運営会社	Anthropic（アンソロピック）	OpenAI（オープン AI）
設立	2021 年（元 OpenAI の研究者が創業）	2015 年
最新モデル（2026 年 4 月）	Claude Opus 4.6、Sonnet 4.6	GPT-5.4、o3、o4
公式サイト	claude.ai	chatgpt.com
日本語対応	あり	あり
コンテキストウィンドウ	最大 100 万トークン、標準 20 万トークン	最大 100 万トークン（GPT-5.4）、標準 27.2 万トークン
画像生成	なし	あり（DALL-E 内蔵）
音声会話	あり（ベータ、全プラン利用可）	あり（Advanced Voice Mode）
Web 検索	あり	あり

9.2.1 Anthropic とはどんな会社か

Anthropic は、もともと OpenAI に在籍していた研究者たちが「より安全な AI を作りたい」という理念で 2021 年に設立した会社です。AI の安全性研究に注力しており、Claude はその研究成果を反映した AI アシスタントです。Google など大手テック企業からの投資を受け、急成長しています。

9.2.2 OpenAI とはどんな会社か

OpenAI は、2015 年にイーロン・マスクらが設立した AI 企業で、世界で最も広く使われている AI アシスタント「ChatGPT」を提供しています。Microsoft の多額の投資を受け、ChatGPT はリリースからわずか 2 ヶ月で月間アクティブユーザー 1 億人を突破した実績があります。

9.3 料金プランの比較

2026 年 4 月時点の料金を比較します。Claude の料金はドル建てのため、1 ドル＝約 150 円で換算しています（実際の請求額は為替レートにより変わります）。

9.3.1 個人向けプランの比較

プラン	Claude	ChatGPT
無料プラン	あり（制限付き）	あり（制限付き）
ミドルプラン	Pro: \$20/月（約 3,000 円）	Plus: \$20/月（約 3,000 円）
ハイプラン	Max 5x: \$100/月（約 15,000 円）	Pro: \$200/月（約 30,000 円）
最上位プラン	Max 20x: \$200/月（約 30,000 円）	—
年払い割引	Pro のみ年払いあり（\$17/月相当）	なし

9.3.2 チーム・法人向けプランの比較

プラン	Claude	ChatGPT
チームプラン	Team Standard: \$25/席/月（最小 5 席）	Business: \$25/席/月（最小 2 席）
チーム上位プラン	Team Premium: \$125/席/月	—
法人プラン	Enterprise（要問い合わせ）	Enterprise（要問い合わせ）

9.3.3 無料プランでどこまで使える？

機能	Claude Free	ChatGPT Free
基本的なチャット	あり	あり
最新モデル	Sonnet 4.6（一部制限）	GPT-5.3（制限付き）
画像アップロード	あり	あり
Web 検索	あり	あり
画像生成	なし	制限あり（DALL-E）
音声会話	あり（ベータ）	制限あり
1 日の利用上限	約 15～40 メッセージ（5 時間ごと）	制限あり（メッセージ数は非公開）



注釈:

2026 年 2 月に Claude の無料プランが大幅に拡充されました。Projects（プロジェクト管理）、Artifacts（成果物の作成・保存）、外部サービス連携（50 以上のアプリ）が無料ユーザーにも開放されています。

つまづきポイント:「無料プランでどっちが使える量が多い？」という質問をよく受けます。どちらも無料プランでは制限がありますが、実際の使い心地は試してみないとわかりません。まずは両方の無料プランを試してから、有料プランへの移行を検討することをおすすめします。

9.4 性能・得意分野の比較

9.4.1 コーディング（プログラミング）

比較軸	Claude	ChatGPT
コード品質の傾向	複雑なりファクタリング・長いコードに強い	幅広い言語・フレームワークに素早く対応
コードの品質	高い（細部への配慮が丁寧）	高い（速度・実用性重視）
専用ツール	Claude Code（ターミナル統合）	Codex
長いコードの処理	最大 100 万トークンで全コードベースを読む	最大 100 万トークン（GPT-5.4）
開発者の評判	コーディング用途で選ばれることが多い（Zapier 調査）	実用的で速い

コーディングではどちらも高い水準にあります。Claude は複雑なリファクタリングや長いコードの一括処理に強く、ChatGPT は幅広い言語・フレームワークへの素早い対応が得意です。各社がベンチマークスコアを公表していますが、テスト条件が異なるため単純な数値比較は難しいのが実情です（詳しくは [AI チャット比較](#)（ページ 75）の「ベンチマークとの付き合い方」を参照）。

9.4.2 文章作成・ライティング

比較軸	Claude	ChatGPT
文体の自然さ	高い（変化に富んだ文章）	標準的（クリーンだが定型的）
長文の理解	最大 100 万トークン対応	最大 100 万トークン対応（GPT-5.4）
創作文章	独創的・文体が豊か	指示に忠実・実用的
ビジネス文章	適切なトーン・構成力に強み	実用的で速い
日本語の品質	自然で違和感が少ない	自然だが時折不自然な表現

文章作成では、複数の比較レビュー（[Zapier](#)、[SurePrompts](#)）で Claude の文章品質が高く評価されています。特に「文章全体のトーンを揃えてほしい」「長い文書を一度に処理したい」という場合に強さを発揮します。

9.4.3 画像生成

比較軸	Claude	ChatGPT
画像生成機能	なし	あり（DALL-E 3 内蔵）
画像の読み取り	あり（写真・図表の分析）	あり（写真・図表の分析）

画像生成では ChatGPT が圧倒的に有利です。Claude は画像を「読む（分析する）」ことはできますが、「作る」ことはできません。マーケティング素材、イラスト、ビジュアルモックアップが必要なら ChatGPT を選ぶべきです。

Claude で画像生成をしたい場合は、Midjourney や Adobe Firefly などの外部ツールとの組み合わせが必要です。

9.4.4 音声会話

比較軸	Claude	ChatGPT
音声会話機能	あり（ベータ、全プラン対応）	あり（Advanced Voice Mode）
通話感覚での対話	可能（英語中心）	可能（多言語対応）

音声会話は ChatGPT がやや有利です。ChatGPT の Advanced Voice Mode は多言語対応で完成度が高く、語学学習やハンズフリー利用には最適です。Claude も音声モードをベータ提供中ですが、現時点では英語中心の対応です。

9.4.5 推論力・論理的思考

比較軸	Claude	ChatGPT
複雑な数学・論理問題	高いレベル	GPT-o3/o4 が特に強い
長い文脈で一貫性	非常に高い（長いコンテキスト対応）	標準的
「考えてから答える」モード	Claude の思考機能（Extended Thinking）	o1/o3 シリーズ

複雑な推論では両者とも高い水準にあります。ChatGPT には数学・科学に特化した o3/o4 シリーズがあり、Claude には Extended Thinking（じっくり考えるモード）があります。得意な問題のタイプが異なるため、実際に試してみるのが一番です。

9.4.6 エコシステム・ツール連携

比較軸	Claude	ChatGPT
カスタム AI（GPTs など）	Projects	GPTs（数百万個以上）
プラグイン・連携サービス	50 以上のアプリ連携	非常に多数
サードパーティ連携	増加中	豊富
ユーザー数	急成長中	世界最大級（月間数億人）

エコシステムの広さでは ChatGPT が圧倒的です。サードパーティ製の GPTs（カスタム AI）が数百万個以上あり、特定の用途（旅行計画、コードレビュー、料理レシピ等）に特化した AI を選んで使えます。

9.5 ユースケース別おすすめ

9.5.1 プログラミング・開発をしたい

おすすめ: Claude

Claude Code（ターミナル統合型の AI コーディングツール）は、大規模なコードベースの把握、バグの特定、リファクタリングに対応しています。コーディングを日常的にする方にはおすすめです。

詳しくは [Claude Code とは？](#)（ページ 3）を参照してください。

9.5.2 日常的なチャット・情報収集

おすすめ: どちらでも（まず両方試して）

日常的なチャットや簡単な質問には、どちらも十分な品質です。無料プランで両方試して、使い心地が好みのほうを選びましょう。

9.5.3 ビジネス文書の作成

おすすめ: Claude

提案書、報告書、メールの作成など、ビジネス文書の作成では Claude の自然な文体と長文処理能力が活きます。長い参考資料を読み込ませて要約・再構成させる作業も得意です。

9.5.4 画像を使ったクリエイティブ作業

おすすめ: ChatGPT

マーケティング用の画像作成、SNS のビジュアル、イラスト作成などには ChatGPT (DALL-E 3) が最適です。テキストから画像を生成したり、既存の画像を編集したりする作業は ChatGPT の独壇場です。

9.5.5 語学学習・外国語の練習

おすすめ: ChatGPT

音声会話機能を活かした英会話練習や、発音を確認しながらの学習には ChatGPT が適しています。

9.5.6 学術研究・論文作成

おすすめ: Claude

長い論文や文献を一度に読み込ませて分析させる作業では、Claude の 100 万トークンのコンテキストウィンドウが圧倒的に有利です。文献のまとめ、論理的な構成の検討、引用の整理に活用できます。

9.5.7 チームでの導入

おすすめ: 用途次第

チーム規模と用途によって異なります。コーディングがメインなら Claude Team (最小 5 席)、幅広い用途なら ChatGPT Business (最小 2 席) が選択肢です。Enterprise プランでは SSO やセキュリティ設定が利用できます。

9.6 併用のすすめ

Claude と ChatGPT は「どちらか一方」ではなく、用途に応じて使い分けるのが最もコスパのよい使い方です。

実際に両方の有料プランを使い続けているパワーユーザーも多く、以下のような使い分けがよく紹介されています。

作業	使うサービス
コードのレビュー・デバッグ	Claude
画像入りのプレゼン素材作成	ChatGPT
長い文書の要約・分析	Claude

[次のページに続く](#)

表 12 – 前のページからの続き

作業	使うサービス
SNS 用のビジュアル作成	ChatGPT
ビジネスメール・報告書の作成	Claude
音声で調べたいとき	ChatGPT



注釈:

両方の無料プランを使うだけなら費用は**ゼロ**です。まずは両方を試して、どちらが自分の仕事スタイルに合っているか確認してから、有料プランへの移行を検討するのが賢い順序です。

9.7 よくある質問

9.7.1 ChatGPT から Claude に乗り換えられますか？

乗り換えというよりも、別のサービスとして登録して使うものです。ChatGPT のアカウントを解約しなくても、Claude のアカウントを新たに作成できます。両方を無料プランで使い続けることも可能です。

9.7.2 データの安全性（プライバシー）に違いはありますか？

どちらも企業ユーザー向けのプランではデータの保護に力を入れています。

- **Claude:** デフォルトでは会話履歴が AI のトレーニングに使用されることがあります。設定で無効化可能です。Enterprise プランではデータのトレーニング利用が行われません
- **ChatGPT:** 同様に、無料・Plus プランでは学習利用される可能性があり、設定でオプトアウト可能。Business/Enterprise では学習利用なし

機密情報を扱うビジネス用途では、有料の Business/Team 以上のプランを使い、プライバシー設定を確認することをおすすめします。

9.7.3 日本語の品質に違いはありますか？

どちらも高品質な日本語に対応しています。比較レビュー（[SurePrompts](#)、[nxcodes](#)）では、Claude の日本語は自然で読みやすいと評価される傾向があります。ただし、個人の好みや用途によって感じ方は異なります。

9.7.4 無料プランから有料プランに移行したら過去の会話は引き継がれますか？

はい、どちらのサービスも同じアカウントでアップグレードするため、過去の会話履歴はそのまま引き継がれます。

9.7.5 どちらのほうの利用者数が多いですか？

ChatGPT のほうが圧倒的に多いです。ChatGPT は月間数億人のユーザーを持ち、世界で最も使われている AI サービスです。Claude は急成長中ですが、ユーザー数ではまだ ChatGPT に及びません。ただし、ユーザー数の多さは使いやすさの優劣とは別の話です。

9.7.6 ChatGPT Plus と Claude Pro、どちらがコスパがよいですか？

どちらも 月額 \$20 (約 3,000 円) で同じ価格帯です。コーディングや長文処理を中心に使うなら Claude Pro、画像生成や音声機能、多様なツール連携を重視するなら ChatGPT Plus がコスパよく使えます。

9.8 まとめ

Claude と ChatGPT の比較をまとめると：

Claude が向いている人：

- コードを書く・レビューする仕事が多い
- 長い文書を読み込ませて分析・要約させたい
- 質の高いビジネス文書を素早く作りたい
- データや論理的な推論を重視する作業が多い

ChatGPT が向いている人：

- 画像を生成・編集したい
- 音声で AI と会話したい
- 特定用途に特化した GPTs を使いたい
- プラグインやサードパーティツールとの連携を重視する

どちらも使いたい人：

- まずは無料プランで両方試してみる
- 用途に応じて使い分けることで最大の効果が得られる

2026 年の現時点では、どちらも世界トップクラスの AI アシスタントであり、得意分野が異なります。「Claude か ChatGPT か」ではなく「Claude と ChatGPT をどう使い分けるか」という視点で考えると、両方のよさを最大限に引き出せます。

9.9 次のステップ

- [Claude とは？](#) (ページ 3) — Claude の基本的な特徴と製品ラインナップを確認する
- [Claude の料金プラン比較](#) (ページ 15) — Claude の Free・Pro・Max・Team の違いを詳しく確認する
- [Claude の料金を日本円で解説](#) (ページ 29) — 日本円での費用感・支払い方法を確認する
- [アカウントの作成と初期設定](#) (ページ 9) — Claude の無料アカウントを作成して試してみる
- [Claude Web 版の使い方ガイド](#) (ページ 89) — ブラウザから Claude を使い始める

第10章 Claude と Gemini の違いを比較【2026年最新】

最終更新: 2026-04-07

「Claude と Gemini、どっちを使えばいいの?」「Gemini は Google のサービスと連携できるって聞いたけど…」「コーディングに使うならどちらが優れている?」

こうした疑問を持っている方のために、2026 年 4 月時点の情報をもとに Claude と Gemini を公正に比較します。どちらのサービスにも強みと弱みがあり、用途によって最適な選択肢は変わります。



警告:

この記事の料金・機能情報は **2026 年 4 月時点** のものです。AI サービスの料金や機能は頻繁に変わります。最新情報は各サービスの公式サイト (claude.ai・gemini.google.com) でご確認ください。

10.1 結論 — 忙しい人向けの要約

まず結論から。どちらを選ぶかは「何に使うか」によります。

用途	おすすめ
コーディング・プログラミング	Claude (特に複雑なリファクタリング)
長文の文章作成・編集	Claude (文体の一貫性が高い)
Google Workspace との連携	Gemini (Gmail・Docs・Drive と深く統合)
動画・音声の分析	Gemini (ネイティブ対応)
リアルタイム情報の検索	Gemini (Google 検索との連携が強い)
長大なドキュメントの一括処理	Gemini (最大 100 万トークンのコンテキスト)
無料で試したい	両方とも無料プランあり (Gemini の無料枠がやや充実)
月額コストを抑えたい	有料プランはほぼ同価格 (Claude Pro \$20、Google AI Pro \$19.99)

どちらか 1 つを選ぶ必要はありません。後述する「併用のすすめ」も参考にしてください。

10.2 基本情報の比較

まず、両サービスの基本的な情報を整理します。

	Claude	Gemini
提供会社	Anthropic（米・サンフランシスコ）	Google DeepMind（米・マウンテンビュー）
設立・背景	2021 年設立。OpenAI 元メンバーが AI 安全性を重視して創業	Google の研究部門 DeepMind が開発。検索・クラウドの巨人
最新フラッグシップモデル	Claude Opus 4.6（高性能）、Claude Sonnet 4.6（バランス型）	Gemini 3.1 Pro（高性能）、Gemini 2.5 Flash（高速・低コスト）
コンテキストウィンドウ	最大 200,000 トークン（約 15 万字相当）	最大 1,000,000 トークン（約 75 万字相当）
日本語対応	対応（自然な日本語）	対応（Google の日本語データを活用）
アプリ	Web・iOS・Android・デスクトップ	Web・iOS・Android
モバイルアシスタント連携	なし	Android デバイスのアシスタントとして統合

10.2.1 コンテキストウィンドウとは

「コンテキストウィンドウ」とは、AI が一度に読み込める情報量のことです。トークンは AI が扱うデータの単位で、日本語の場合、1 文字が 1〜2 トークン程度になります。

- **Claude:** 200,000 トークン ≒ 文庫本約 1〜2 冊分
- **Gemini:** 1,000,000 トークン ≒ 文庫本約 5〜8 冊分

Gemini のコンテキストウィンドウは Claude の 5 倍。「大量のドキュメントをまとめて分析したい」「長い動画の内容を理解させたい」という場面では Gemini が有利です。

10.3 料金プランの比較

10.3.1 無料プラン

どちらのサービスも無料で使い始められます。

	Claude Free	Gemini（無料）
月額	無料	無料
使えるモデル	Claude Sonnet・Haiku	Gemini 2.0 Flash（軽量モデル）
利用上限	5 時間あたり 15〜40 メッセージ程度	1 日のクエリ数に制限あり
Google サービス連携	Gmail 連携（Connector 経由）	Gmail・Drive・Docs・YouTube（無料でも一部利用可）
ストレージ	なし	なし（Google One ストレージ分離）



注釈:

Gemini の無料プランは Google アカウントを持っていれば即座に利用できます。Gmail や Google 検索と軽く連携する程度なら、無料プランでも十分に活用できます。

10.3.2 有料プラン比較

	Claude Pro	Google AI Pro
月額 (USD)	\$20	\$19.99
月額 (円換算)	約 3,000 円	約 3,000 円 (日本では 2,900 円)
使えるモデル	Claude Opus 4.6・Sonnet・Haiku	Gemini 3.1 Pro (最新フラッグシップ)
利用量	無料の約 5 倍	「Higher」上限 (具体的な数は非公開)
追加ストレージ	なし	Google One 5TB
Google Workspace 連携	基本連携	Gmail・Drive・Docs・Slides・Sheets に深く統合
コーディングツール	Claude Code 利用可	Google の Gemini Code Assist (IDE 統合)
NotebookLM 強化	なし	ノートブック 500 件、ソース 300 件/ノート

Google AI Pro の旧名称: Google は 2025 年末に「Google One AI Premium」を「Google AI Pro」に改称しました。名前は変わりましたが、内容は同等のプランです。

10.3.3 パワーユーザー向けの上位プラン

より多くの利用量が必要な場合、両サービスともに上位プランを提供しています。

	Claude Max (5x)	Claude Max (20x)	Google AI Ultra
月額	\$100	\$200	\$249.99 (約 37,000 円)
特徴	Pro の 5 倍の利用量	Pro の 20 倍の利用量	Gemini 3.1 Pro・Veo 3.1 (動画生成)・Deep Think モード
向いている人	頻繁に上限に達する開発者	ほぼ一日中使う開発者	Google のすべての AI 機能を使い倒したい人

Claude の Max プランについて詳しくは [Claude の料金プラン比較](#) (ページ 15) をご覧ください。

10.4 性能・得意分野の比較

10.4.1 コーディング (プログラミング)

	Claude	Gemini
コード品質の傾向	複雑なリファクタリング・指示への忠実な実行に強い	大量のコード処理・Google Cloud 連携に強い
強み	複雑なリファクタリング、指示への忠実な実行、大規模コードベースの理解	複数ファイルにまたがる推論、Google Cloud・Firebase との連携
IDE 統合	Claude Code（ターミナルベース）	Gemini Code Assist（VS Code・JetBrains）

コーディングではどちらも高い水準にあります。各社がベンチマークスコアを公表していますが、テスト条件やバージョンが異なるため単純な数値比較は難しいのが実情です（詳しくは [AI チャット比較](#)（ページ 75）の「ベンチマークとの付き合い方」を参照）。

つまづきポイント：「Gemini で書かせたコードが動かない…」という声もあります。どちらのサービスも完璧ではなく、生成されたコードは必ず動作確認が必要です。特に複雑なロジックが絡む場合はテストを怠らないようにしましょう。

10.4.2 文章作成・編集

	Claude	Gemini
長文の一貫性	高い（文体・トーンを維持しやすい）	やや変動しやすい
構造化された文書	得意	レポートや論文形式が得意
自然な文章表現	自然でこなれた表現	情報は正確だが機械的になりやすい
インターネット情報の参照	Web 検索機能あり	Google 検索との深い連携で最新情報に強い

文章のトーン管理や長編記事の一貫した品質維持では Claude が優れているという評価があります。一方、事実確認が必要な記事やリサーチ系の文書では、Google 検索と連動する Gemini が強みを発揮します。

10.4.3 マルチモーダル（画像・動画・音声の理解）

「マルチモーダル」とは、テキスト以外の情報（画像・動画・音声など）も処理できる能力のことです。

	Claude	Gemini
画像の理解・分析	対応	対応
動画の理解	非対応	対応（最大 3 時間の動画を処理可能）
音声の理解	非対応	対応
画像生成	非対応	Imagen モデルで対応（有料プランで利用量増加）
動画生成	非対応	Veo モデルで対応（Ultra プランで高品質）

マルチモーダルの対応範囲は Gemini が大幅に広いです。動画や音声を扱う必要があるなら Gemini 一択といえます。

10.4.4 推論力・複雑な問題解決

	Claude	Gemini
拡張思考（じっくり考えるモード）	Extended Thinking（Pro 以上で利用可）	Deep Think モード（Ultra プランで利用可）
数学・論理問題	高い精度	同等以上の精度
長文ドキュメントの理解	最大 200K トークン	最大 1M トークン（Gemini が有利）

10.5 Google エコシステムとの統合

ここは Gemini の圧倒的な強みです。Google Workspace を日常的に使っている方にとって、Gemini の統合機能は非常に価値が高いです。

10.5.1 Gemini が対応している Google サービス

サービス	機能
Gmail	メールの要約、返信文の自動生成、検索の AI 概要
Google ドキュメント	文書の下書き・要約・改善提案
Google スプレッドシート	データ分析、数式の生成、グラフの作成提案
Google スライド	プレゼンの構成提案、デザイン改善
Google ドライブ	ファイル横断の検索・要約
Google Meet	会議の要約、アクションアイテム抽出
NotebookLM	アップロードした資料をもとにした Q&A（幻覚（でたらめな情報）なし）
Google Search	検索結果の AI 概要（AI Overviews）
YouTube	動画の要約（Gemini アプリ経由）

10.5.2 Claude の Google Workspace 連携

Claude も Connectors 機能で Google Workspace と連携できますが、Gemini ほどの深い統合ではありません。Claude の Gmail 連携はあくまで「Claude からメールを読み書きする」レベルで、Google Workspace のサイドパネルに常駐するような形では使えません。



注釈:

Google Workspace を会社全体で使っていて、日常業務でドキュメントやスプレッドシートを頻繁に使う場合は、Gemini の方が業務効率化の恩恵を受けやすいです。

10.6 ユースケース別おすすめ

10.6.1 Google Workspace を日常的に使うビジネスパーソン

→ Gemini (Google AI Pro) を優先検討

Gmail の返信文作成、Google ドキュメントの下書き、スプレッドシートのデータ分析など、Google のツール内で AI を活用したい場合、Gemini の統合機能は他の AI サービスでは代替が難しいです。

10.6.2 コーディングやプログラミングを行うエンジニア

→ Claude (Pro 以上) を優先検討

Claude にはターミナルベースの **Claude Code** があり、プロジェクト全体を把握した自律的なコーディングが可能です。Gemini には Gemini Code Assist (VS Code・JetBrains 統合) があります。

10.6.3 長大なドキュメントや資料を一括で処理したい

→ Gemini を優先検討

契約書・研究論文・マニュアルなど、大量のテキストを一度に読み込んで分析したい場合、100 万トークンのコンテキストウィンドウを持つ Gemini が向いています。特に NotebookLM は資料をアップロードするだけで幻覚なしで Q&A できる独自の機能です。

10.6.4 高品質な長文ライティング

→ Claude を優先検討

ブログ記事・レポート・提案書など、文体の一貫性が重要な長文を書く場合、複数の比較レビュー (**DataCamp**、**lucas8**) で Claude の文章品質が高く評価されています。特に「このトーンで書いて」という指示を忠実に守る精度が高いとされています。

10.6.5 動画・音声コンテンツの分析

→ Gemini 一択

YouTube 動画の要約、音声ファイルの文字起こし・分析など、動画や音声を扱う場合は Claude には未対応のため Gemini を使ってください。

10.6.6 コストを抑えて両方のサービスを試したい

→ 両方の無料プランで試す

Claude の Free プランと Gemini の無料プランはどちらも 0 円で使えます。まず両方試してみて、自分の用途に合った方を選ぶのが一番です。

10.7 併用のすすめ

「Claude か Gemini か」という二択で考える必要はありません。月 3,000 円前後のプランをどちらか一方に絞るとすれば用途に合った方を選ぶべきですが、両方を使い分けることも十分に合理的な選択です。

10.7.1 効果的な使い分けの例

タスク	使うサービス	理由
Gmail の返信文作成	Gemini	Gmail 内で直接使える
ブログ記事・提案書の執筆	Claude	文体の一貫性が高い
動画の内容を理解する	Gemini	動画処理に対応
コードのレビュー・修正	Claude	コーディング精度が高い
大量の PDF を一括要約	Gemini	大きなコンテキストウィンドウ
ニュースや最新情報の調査	Gemini	Google 検索との連携

つまづきポイント:「両方有料プランにすると月 6,000 円になってしまう…」という場合は、まず 1 つを有料プランにして、もう 1 つは無料プランで使い続けるのが現実的です。無料プランでも基本的な用途は十分カバーできます。

10.8 よくある質問

10.8.1 無料で使えるのはどちらですか？

両方とも無料プランがあります。どちらも無料登録だけで基本機能が使えます。Gemini は Google アカウントがあれば追加登録不要なため、Google サービスをすでに使っている方はすぐに使い始められます。

10.8.2 日本語の対応はどちらが良いですか？

両サービスとも日本語に対応しており、自然な日本語でやり取りできます。評価は用途によって異なりますが、比較レビュー ([DataCamp](#)) では文章作成の自然さで Claude が高く評価される傾向があります。Gemini は Google の日本語データを活用しているため、日本のローカル情報の検索に強みがあります。

10.8.3 コンテキストウィンドウの違いはどのくらいですか？

Claude は最大 200,000 トークン（文庫本 1〜2 冊分）、Gemini は最大 1,000,000 トークン（文庫本 5〜8 冊分）です。Gemini の方が 5 倍大きいため、大量のドキュメントを一度に処理したい場合は Gemini が有利です。

10.8.4 Google Workspace を使っているなら Gemini 一択ですか？

必ずしもそうではありません。Google Workspace ユーザーでも、コーディングや高品質な長文ライティングには Claude を使うという選択肢もあります。Gemini の Google 統合機能が「自分の日常業務で本当に使うか」を考えてから判断しましょう。

10.8.5 Claude Code に相当する Gemini の機能はありますか？

Gemini には「Gemini Code Assist」という IDE（コードエディター）に統合されるコーディング支援ツールがあります。VS Code や JetBrains の IDE でサイドバーから使えるため、IDE を使い慣れたエンジニアには使いやすい形です。一方、Claude Code はターミナル（コマンドプロンプト）から動かす形式で、より自律的にコードを書いたり修正したりすることが得意です。

10.8.6 安全性・プライバシーはどうですか？

Anthropic は「AI の安全性を最優先にする企業」として設立され、Claude の応答品質と安全性に注力しています。Google は巨大なデータ処理インフラと実績があります。いずれのサービスも有料プランでは会話データをモデル学習に使わないオプションがあります（詳細は各社の利用規約・プライバシーポリシーを参照）。

10.8.7 日本から利用できますか？

はい、両サービスとも日本から利用できます。Gemini の Google AI Pro プランは日本では月額 2,900 円で提供されており、ドル建てではなく円建て決済が可能です。Claude の料金はドル建てのため、クレジットカードの為替手数料が発生する点に注意してください。詳しくは [Claude の料金を日本円で解説](#)（ページ 29）をご覧ください。

10.9 まとめ

2026 年 4 月時点での Claude と Gemini の比較をまとめます。

Claude が向いている場面:

- コーディング・プログラミング（特に複雑な修正や大規模プロジェクト）
- 高品質な長文ライティング（文体の一貫性重視）
- 構造的な分析・推論
- Claude Code を使った自律的なコーディング作業

Gemini が向いている場面:

- Google Workspace（Gmail・Docs・Drive 等）との深い連携
- 動画・音声の分析・要約
- 大量のドキュメントを一括処理（最大 100 万トークン）
- Google 検索との連携が必要なリサーチ作業
- NotebookLM を使った資料ベースの Q&A

どちらも同じくらい優れている場面:

- 日常的な質問・情報収集
- 簡単な文章作成・翻訳
- データの整理・分類

どちらが「最高」かは用途次第です。まずは両方の無料プランを試してみて、自分の使い方に合った方を有料プランにアップグレードするのが一番賢い選択でしょう。

10.10 次のステップ

- *Claude* の料金プラン比較 (ページ 15) — Free・Pro・Max・Team の詳細を確認する
- *Claude* の料金を日本円で解説 (ページ 29) — 日本円での費用感を把握する
- *Claude* とは？ (ページ 3) — Claude の基本的な特徴を理解する
- *Claude Web* 版の使い方ガイド (ページ 89) — 今すぐ Claude を使い始める

第11章 AI チャット比較【2026年版】 Claude・ChatGPT・Gemini・Copilot

最終更新: 2026-04-07

「ChatGPT と Claude って何が違うの?」「Gemini や Copilot も気になるけど、どれを使えばいいかわからない」

このページでは、2026 年現在の主要 4 サービス（Claude・ChatGPT・Gemini・Copilot）を料金・性能・用途別に徹底比較します。比較表を使って、3 分で自分に合ったサービスが選べるようにまとめました。



警告:

この記事の料金・機能情報は **2026 年 4 月時点** のものです。各サービスの仕様は頻繁に更新されます。最新情報は各公式サイトでご確認ください。

11.1 結論：忙しい人向け一覧比較表

まずは 4 サービスを一表で比較します。「用途が決まっている」という方は、この表だけ見れば OK です。

	Claude	ChatGPT	Gemini	Copilot
運営会社	Anthropic	OpenAI	Google	Microsoft
無料プラン	あり	あり	あり	あり
有料プラン (個人)	\$20/月～ (約 3,000 円～)	\$8/月～ (約 1,200 円～)	\$19.99/月 (約 3,000 円)	無料+Microsoft 365 統合
最新モデル	Claude Opus 4.6 / Sonnet 4.6	GPT-5.4 / GPT-4o	Gemini 3.1 Pro	GPT-5 ベース (マルチモデル)
コーディング	◎ 非常に得意	○ 得意	○ 得意	◎ IDE 統合が強い
文章作成	◎ 非常に自然	○ 得意	○ 得意	○ Word 統合が強い
画像生成	× 非対応	◎ DALL-E 3 搭載	○ Imagen 搭載	◎ Designer 搭載
Google 連携	△ 一部対応	× なし	◎ Gmail/Docs 完全統合	× なし
Microsoft 連携	△ 一部対応	△ 一部対応	× なし	◎ Office 完全統合
長文処理	◎ 200K トークン	◎ 最大 1M トークン (GPT-5.4)	◎ 1M トークン	○ 128K トークン

次のページに続く

表 1 – 前のページからの続き

	Claude	ChatGPT	Gemini	Copilot
おすすめ用途	コーディング・長文・高品質文章	多用途・画像生成・日常使い	Google ユーザー・調査	Office/Windows ユーザー

「どれか 1 つだけ選ぶなら？」

- ・ コーディングや複雑な文章作成が多い → Claude
- ・ 画像生成もしたい、または色々試したい → ChatGPT
- ・ Google Workspace（Gmail・Docs）を日常的に使う → Gemini
- ・ Microsoft Office（Word・Excel）中心に仕事をする → Copilot

11.2 各サービスの概要

11.2.1 Claude（Anthropic）

Claude は、AI 安全性の研究に特化した Anthropic（2021 年設立、米国）が開発した AI アシスタントです。「安全で誠実で有益な AI」を開発理念として掲げており、ハルシネーション（AI が事実でないことを自信を持って語る現象）が少なく、微妙なニュアンスを理解した自然な文章を生成するとされています（Zapier）。

Claude の強み:

- ・ 複数の比較レビューで文章の自然さ・ニュアンスの理解が高く評価されている（Zapier、SurePrompts）
- ・ コーディング能力（特に複雑なコードのデバッグ・リファクタリング）が非常に高い
- ・ 長い文書（200K トークン、約 15 万字相当）を一度に処理できる
- ・ 安全性への配慮が高く、業務での利用に安心感がある

Claude の弱み:

- ・ 画像生成機能がない（画像の分析・説明はできる）
- ・ Google・Microsoft エコシステムとの統合度は低い

詳しくは [Claude とは？](#)（ページ 3）をご覧ください。

11.2.2 ChatGPT（OpenAI）

ChatGPT は、AI 研究機関 OpenAI が開発した世界で最もユーザー数の多い AI チャットサービスです（2026 年 2 月時点で週間アクティブユーザー 9 億人超）。2022 年 11 月のリリース以来、世界中に AI チャットを普及させたサービスです。

ChatGPT の強み:

- ・ DALL-E 3 による高品質な画像生成が可能
- ・ 幅広い用途に対応する多機能性（GPT-5.4、データ分析、Web 検索等）
- ・ ユーザー数が最多のため、使い方の情報が豊富

- GPT を活用したカスタム AI (GPTs) のエコシステムが充実
- 段階的な料金プランで入門しやすい (月 \$8 の「Go」プランあり)

ChatGPT の弱み:

- Google・Microsoft エコシステムとの統合は薄い
- サービスの更新頻度が高く、機能の変更が多い

11.2.3 Gemini (Google)

Gemini は、Google が開発した AI アシスタントです。Gmail・Google Docs・Google スプレッドシートなど Google Workspace との深い統合が最大の特徴です。2025 年の「Google I/O」以降、「**Google AI Pro**」というブランド名で提供されています。

Gemini の強み:

- Gmail、Docs、Sheets、Meet など Google Workspace に直接統合
- 最大 100 万トークン (Gemini 3.1 Pro) という業界最大規模のコンテキストウィンドウ
- Google 検索との連携による高い情報収集能力
- Imagen による画像生成機能

Gemini の弱み:

- Google Workspace 以外の環境では、連携の恩恵が少ない
- 文章の自然さは Claude に劣るという評価が多い

11.2.4 Copilot (Microsoft)

Microsoft Copilot は、OpenAI の GPT-5 シリーズを基盤とした Microsoft 製の AI アシスタントです。Word・Excel・PowerPoint・Outlook・Teams など Microsoft 365 アプリに深く統合されており、**既存の仕事のフローの中に AI を組み込む**というコンセプトが特徴です。2026 年からは Anthropic の Claude も統合し、マルチモデル戦略を採用しています ([Microsoft 公式ブログ](#))。

Copilot の強み:

- Word、Excel、PowerPoint などの Office アプリに直接統合
- Designer による画像生成が無料で使える
- Windows 11 に標準搭載されており、追加インストール不要
- GPT-5.4 Thinking (推論モデル) が Microsoft 365 Copilot で利用可能 (公式発表)
- Claude も Researcher 機能経由で利用可能 (マルチモデル対応)

Copilot の弱み:

- Microsoft 365 を使っていないユーザーには機能が限定される
- AI 機能のフル活用には Microsoft 365 Premium (年額 \$199.99) が必要

11.3 料金プランの横断比較

2026 年 4 月時点の料金情報です。すべてドル建てのため、為替レートによって日本円での支払額は変動します（1 ドル＝約 150 円換算）。

11.3.1 個人向け無料プラン比較

	Claude Free	ChatGPT Free	Gemini Free	Copilot（無料）
料金	無料	無料	無料	無料
主なモデル	Claude Sonnet 4.6	GPT-4o（制限あり）	Gemini 2.0 Flash	GPT-5 ベース
メッセージ上限	5 時間あたり約 15 ～40 件	1 日約数十件（変動）	比較的寛大	比較的寛大
画像生成	×	×	△（制限あり）	○（Designer、1 日 15 枚程度）
Web 検索	○	○	○	○
特記事項	—	—	Google Workspace 連携あり	Windows 11 標準搭載



初心者へのアドバイス: まずは全サービスとも無料プランで試してみましょう。実際に使ってみると「このサービスが自分には合う」という感覚がつかめます。

注釈:

11.3.2 個人向け有料プラン比較

	Claude Pro	ChatGPT Plus	Google AI Pro	Microsoft 365 Premium
月額(ドル)	\$20/月	\$20/月	\$19.99/月	年額 \$199.99（約 \$17/月）
月額(円目安)	約 3,000 円（税込約 3,300 円）	約 3,000 円	約 3,000 円	年額約 30,000 円
主なモデル	Claude Opus 4.6, Sonnet 4.6	GPT-5.4, GPT-4o	Gemini 3.1 Pro	GPT-5.4 Thinking, Claude 対応
画像生成	×	◎ DALL-E 3（高品質）	○ Imagen 搭載	◎ DALL-E 3
特典	利用量 5 倍、Claude Code 利用可	高メッセージ上限、高度データ分析	Google Workspace 連携、5TB ストレージ	Office アプリ統合
年払い割引	あり（\$17/月相当）	不明	あり（1 ヶ月無料トライアル）	Microsoft 365 年払いで割引



注釈:

ChatGPT の新プランについて: 2026 年から ChatGPT は「Go (\$8/月)」「Plus (\$20/月)」「Pro (\$200/月)」など複数の段階的なプランを提供しています。手軽に始めたい方には \$8/月の「Go」プランも選択肢になります。

11.3.3 ビジネス・チーム向けプラン比較

	Claude Team	ChatGPT Business	Gemini Business	Microsoft 365 Copilot
月額 (1 人あたり)	\$25~\$125/月	\$25/月	要確認	\$18~\$30/月
最低人数	5 名	2 名から	1 名から	1 名から
管理機能	○	○	○	◎
SSO 対応	○	○	○	○
特記事項	シート種別あり (Standard/Premium)	GPT-5.4 含む全モデル	Google Workspace 追加	Office 全アプリ統合

11.4 性能比較

実際の使用場面ごとに、各サービスの強弱を比較します。

11.4.1 コーディング・プログラミング

	Claude	ChatGPT	Gemini	Copilot
評価	◎	○	○	◎ (IDE 統合)
特徴	複雑な問題解決・デバッグに強い。コードの意図の理解が深い	幅広い言語・ライブラリに対応。実用的な解を素早く出す	速度が速い。大量のコードを一度に処理できる	VS Code・GitHub Copilot との統合が強力

Claude Code (ターミナルで動く AI コーディングツール) を使うと、プロジェクト全体を理解したうえでコードを編集できます。

つまづきポイント: 「ChatGPT でコードを書いてもらったら動かなかった」という方が多くいます。どのサービスも複雑なコードでは間違いが混じることがあります。生成されたコードは必ず動作確認してから使いましょう。

11.4.2 文章作成・翻訳

	Claude	ChatGPT	Gemini	Copilot
評価	◎	○	○	○ (Word 統合)
特徴	人間らしい自然な文章。トーンや目的の理解が深い	多用途。ビジネス文書から詩まで対応	Google ドキュメント内で直接編集可	Word 内での文章改善・補完に強い

Claude の文章は「AI っぽさ」が少なく自然だと複数の比較レビュー ([Zapier](#)、[SurePrompts](#)) で評価されています。長文のレポートや複雑なビジネス文書の作成には特に向いています。

11.4.3 データ分析・表計算

	Claude	ChatGPT	Gemini	Copilot
評価	○	◎	○	◎ (Excel 統合)
特徴	ファイルの分析・数値の読み解きが得意	高度データ分析 (Code Interpreter) で強力	Google スプレッドシートに直接統合	Excel 内での数式生成・データ整理が直感的

Excel を使っている方には Copilot、Google スプレッドシートを使っている方には Gemini が自然に統合されます。

11.4.4 画像生成

	Claude	ChatGPT	Gemini	Copilot
評価	×	◎	○	◎
特徴	非対応 (画像の分析・説明は可能)	DALL-E 3 による高品質な画像生成	Imagen による画像生成	Designer (DALL-E 3) が無料で利用可能

画像生成を重視する場合は、Claude 以外の 3 サービスのいずれかを選ぶのが基本です。

11.4.5 長文・大量データの処理

	Claude	ChatGPT	Gemini	Copilot
コンテキスト長	200K トークン (約 15 万字)	最大 1M トークン (GPT-5.4)	1M トークン (約 75 万字)	128K トークン (約 9.6 万字)
評価	◎	○	◎ (最大)	○

コンテキストウィンドウとは、AI が一度に読み込める文章の量のことです。数値が大きいほど、長い文書を丸ごと渡して処理できます。Gemini は最大 100 万トークンと最大規模を誇りますが、Claude の 200K も書籍 1 冊分以上を処理できる十分な容量です。

11.5 ベンチマークとの付き合い方

AI サービスを比較するとき、「SWE-bench」「MMLU」「HumanEval」などのベンチマークスコアがよく引き合いに出されます。しかし、ベンチマークの数値だけでサービスを選ぶのはおすすめしません。

ベンチマークを鵜呑みにできない理由:

- **テスト条件が統一されていない** — 同じベンチマーク名でも「Verified」「Pro」「Lite」などバリエーションがあり、各社が異なる条件（試行回数、プロンプト調整の有無等）でスコアを出している
- **各社が有利なベンチマークを選んで発表する** — 自社モデルが高スコアを出したベンチマークを大きく取り上げ、そうでないものは触れない傾向がある
- **スコアが拮抗している** — 2026 年 4 月時点で、主要モデルのコーディングベンチマークは 80% 前後に集中しており、数ポイントの差で「圧勝」「大差」とは言えない
- **ベンチマークと実際の使用感は別** — テスト問題を解く能力と、あなたの日常業務で役立つかは別の話。ベンチマーク 1 位のモデルが、あなたの用途に最適とは限らない

では、何を基準に選べばよいか？

1. **自分の用途で実際に試す** — 無料プランで同じ質問・同じタスクを各サービスに投げて、回答の質を比較するのが最も確実
2. **機能差で選ぶ** — 画像生成の有無、Google/Microsoft との統合、コンテキストウィンドウの大きさなど、ベンチマークでは測れない機能差の方が実用的な判断基準になる
3. **更新頻度を意識する** — AI モデルは数ヶ月単位で入れ替わる。今日のスコアは半年後には古い情報になっている

つまづきポイント: 「〇〇のベンチマークで1位だからこのサービスが最強」という記事を見かけたら、どのバージョンのベンチマークか、テスト条件は何か、他のベンチマークではどうかを確認しましょう。1つの数値でAIの総合力は測れません。

11.6 用途別おすすめランキング

11.6.1 プログラミング・開発

1 位: Claude — コードの品質と複雑な問題の解決力が最も高い。Claude Code を使えばプロジェクト全体に対してインタラクティブに作業できる

2 位: Copilot (GitHub Copilot) — VS Code や JetBrains などの IDE に直接統合。コードを書きながらリアルタイムで補完してくれる

3 位: ChatGPT — 幅広い言語・フレームワークに対応。実用的なコードを素早く生成する

11.6.2 ビジネス文書作成

1 位: Claude — 最も自然で読みやすい文章を生成。会議の議事録、報告書、プレゼン資料の草案に最適

2 位: Copilot — Word 内で直接文章を補完・改善できる。既存の Office ワークフローを崩さずに使える

3 位: ChatGPT — 多様な文書形式に対応。テンプレートの生成が得意

11.6.3 日常的な質問・情報収集

1 位: ChatGPT — 最もユーザー数が多く、使い方の情報も豊富。汎用性の高さが魅力

1 位: Gemini — Google 検索の情報を取り込みやすく、最新情報の確認に強い

3 位: Claude — 正確で詳細な説明が得意。複雑な質問の回答品質が高い

11.6.4 学術研究・論文作業

1 位: Claude — 長文の論文を丸ごと読み込んで要約・分析できる。文章の品質・正確さが高い

2 位: Gemini — 1M トークンという最大規模のコンテキストで大量の資料を処理可能

3 位: ChatGPT — 多様な分野の知識と、数式・データ分析ツールへの対応

11.6.5 クリエイティブ作業（小説・イラスト・デザイン）

1 位: ChatGPT — DALL-E 3 による画像生成と、創作的な文章作成の両方をカバー

2 位: Copilot — Designer（DALL-E 3）が無料で使える。Windows ユーザーには最もアクセスしやすい

3 位: Gemini — Imagen による画像生成と、Google フォトとの連携

11.6.6 Google Workspace ユーザー

1 位: Gemini（Google AI Pro） — Gmail、Docs、Sheets、Meet に完全統合。追加ツール不要で AI を使えるようになる

11.6.7 Microsoft 365 ユーザー

1 位: Copilot — Word、Excel、PowerPoint、Teams に直接統合。既存の業務フローに AI が入り込む

11.7 無料で始めるならどれ？

各サービスとも無料プランが用意されています。「まず試してみたい」という方向けに、無料プランの使い勝手を比較します。

サービス	無料の充実度	向いている方
Claude	○ 毎日数回程度ならストレスなく使える	文章作成・要約・翻訳を試したい方
ChatGPT	◎ GPT-4o に一定回数アクセスできる	何でも試してみたい方、画像生成も気になる方
Gemini	◎ 無料枠が比較的寛大	Google Workspace ユーザー
Copilot	◎ Windows 11 に標準搭載、すぐ使える	Windows ユーザー、Office ユーザー

11.7.1 初心者への具体的なアドバイス

「とにかく手軽に始めたい」：Windows を使っているなら Copilot が最も入り口が低いです。Windows 11 のタスクバーからすぐに使えます。

「色々試したい」：ChatGPT の無料プランが最も汎用性が高く、情報も豊富です。

「仕事で使いたい」：Claude の無料プランを試してみましょう。ビジネス文書の作成や要約に強みがあります。Google Workspace を使っているなら Gemini も試してみる価値があります。

つまづきポイント：「無料プランを試したら反応が遅すぎて使えなかった」という声があります。無料プランは混雑時に速度が落ちることがあります。特に日本時間の午後（米国東海岸の昼間と重なる時間帯）は込み合いやすいです。朝や夜の時間帯だとスムーズに使えることが多いです。

11.8 よくある質問

11.8.1 結局どれが一番いいの？

「最高の AI サービス」は存在せず、**用途と自分のワークフロー次第**です。

- コーディングや高品質な長文作成 → Claude
- 画像生成や多用途 → ChatGPT
- Google Workspace 活用 → Gemini
- Office 活用 → Copilot

迷うなら、まず ChatGPT か Claude の無料プランを両方試して、感覚が合う方を使い続けるのがおすすめです。

11.8.2 複数のサービスを使い分けるべき？

使い分けは十分アリです。例えば「文章の草案は Claude で作り、挿入する画像は ChatGPT（DALL-E 3）で生成する」という使い方をしている方も多くいます。

慣れてきたら、以下のような使い分けが効率的です：

- 日常の質問・情報収集 → Copilot または Gemini（無料で十分）
- 重要な文書作成 → Claude（品質を重視）
- 画像生成が必要なとき → ChatGPT か Copilot

11.8.3 セキュリティやプライバシーが心配

いずれのサービスも無料プランでは会話内容が AI の改善に使われる可能性があります。**機密情報や個人情報を含む内容は、無料プランで入力しないことをおすすめします。**

有料プランや法人向けプランでは、データの学習利用をオプトアウト（拒否）できる機能が用意されています。

- **Claude**: Pro プランから会話内容の学習利用をオフにできる

- **ChatGPT:** 設定でチャット履歴の学習をオフにできる（無料・有料両方）
- **Gemini:** Google の設定からデータ管理が可能
- **Copilot:** Microsoft 365 の設定でコントロール可能

11.8.4 日本語の対応はどうか？

4 サービスとも日本語に対応しており、日本語での会話に問題ありません。

ただし、日本語対応の品質には多少差があります：

- **Claude:** 日本語の自然さが高い。ニュアンスのある表現も理解できる
- **ChatGPT:** 日本語対応が充実。日本語の情報も豊富
- **Gemini:** 日本語に対応。Google 検索との連携で日本語情報へのアクセスが強い
- **Copilot:** 日本語に対応。Microsoft の日本法人が存在するため、日本語リソースも充実

11.9 まとめ

4 サービスの特徴を改めて整理します。

サービス	最大の強み	こんな人におすすめ
Claude	文章品質・コーディング力・長文処理	文章作成が多いビジネスパーソン、開発者
ChatGPT	多機能・画像生成・ユーザー数が多い	何でも使いたい、情報収集したい
Gemini	Google Workspace 統合・超長文処理	Google ユーザー、調査・リサーチ重視
Copilot	Microsoft 365 統合・無料での画像生成	Office ユーザー、Windows ユーザー

どれを選んでもハズレはありません。4 サービスとも無料プランがあるので、まず試してみることが最善策です。

11.9.1 各サービスの詳細比較

Claude と他のサービスを 2 サービス間で深掘りした比較記事もあります：

- [Claude と ChatGPT の違いを徹底比較](#) (ページ 57)
- [Claude と Gemini の違いを比較](#) (ページ 65)

11.10 次のステップ

- [Claude とは？](#) (ページ 3) — Claude の概要と製品ラインナップを知る
- [Claude の料金プラン比較](#) (ページ 15) — Claude の Free・Pro・Max の違いを詳しく比較する
- [アカウントの作成と初期設定](#) (ページ 9) — Claude の無料プランを今すぐ始める
- [Claude と ChatGPT の違いを徹底比較](#) (ページ 57) — 2 サービスの詳細な差を確認する

- *Claude* と *Gemini* の違いを比較 (ページ 65) — Google Gemini との詳細な比較

第II部

製品ガイド

第12章 Claude Web (claude.ai) の使い方

Claude Web は、Anthropic が提供するブラウザベースの AI チャットサービスです。claude.ai にアクセスするだけで、誰でもすぐに AI との会話を始められます。ソフトウェアのインストールは不要で、パソコンでもスマートフォンでも利用できます。

この記事では、アカウント作成から主要機能の活用方法まで、Claude Web の使い方を一通り解説します。

12.1 Claude Web とは

Claude Web は、ブラウザ上で Claude と会話できるサービスです。文章の作成、質問への回答、データの分析、コードの生成など、さまざまな作業を自然な言葉で依頼できます。

無料プラン (Free) でも基本的な機能は使えるので、まずは気軽に試してみましょう。

12.1.1 アカウントの作成とログイン

1. ブラウザで claude.ai にアクセスします
2. 「Sign up」をクリックして新規登録を始めます
3. メールアドレス、または Google / Apple アカウントで登録できます
4. メールアドレスで登録した場合は、届いた確認メールのリンクをクリックして認証を完了します
5. ログインすると、すぐにチャット画面が表示されます

つまづきポイント: ログインできない場合は、ブラウザの Cookie やキャッシュをクリアしてみてください。また、VPN を使っている場合は一時的にオフにすると解決することがあります。

12.1.2 料金プラン

Claude Web には複数のプランがあります。

プラン	月額料金	主な特徴
Free	無料	基本的な会話が可能。1日あたりのメッセージ数に制限あり
Pro	\$20	メッセージ上限が大幅に増加。Opus モデルが利用可能
Max 5x	\$100	Pro の 5 倍の利用量。Claude Code も利用可能
Max 20x	\$200	Pro の 20 倍の利用量。最優先アクセス

Free プランの制限事項: 1日あたり約 30~100 メッセージまで送信できます (会話の長さや添付ファイルの有無により変動します)。制限に達した場合は、数時間待つとリセットされます。また、ピーク時間帯 (平

日の日中など) はレスポンスが遅くなったり、一時的にアクセスが制限されることがあります。

12.2 基本の会話

12.2.1 メッセージの送り方

Claude との会話はとてもシンプルです。

1. 画面下部のテキスト入力欄にメッセージを入力します
2. Enter キー (または送信ボタン) を押すと、メッセージが送信されます
3. Claude が数秒で回答を生成します

普段の会話のように、自然な言葉で話しかければ OK です。例えば次のようなことが聞けます。

- 「来週の社内ミーティングのアジェンダを作って」
- 「この文章をもっとわかりやすくリライトして」
- 「売上データの傾向を分析して」
- 「英語のメールを日本語に翻訳して」

つまづきポイント: Claude の回答が期待と違う場合は、より具体的に指示を出してみましょう。「わかりやすく」ではなく「中学生でもわかるように」、「短くして」ではなく「200 文字以内で」のように、条件を明確にすると精度が上がります。

12.2.2 画像・ファイルの添付

Claude は文字だけでなく、画像やファイルを読み取って回答することもできます。

- **画像:** スクリーンショットや写真をドラッグ&ドロップ、またはクリップアイコンから添付できます。「この画像の内容を説明して」「このグラフから読み取れることは？」といった質問ができます
- **ドキュメント:** PDF、Word、テキストファイルなどを添付して、内容の要約や分析を依頼できます
- **データファイル:** CSV や Excel ファイルを添付して、データの分析やグラフの作成を依頼できます

12.2.3 会話の管理

Claude Web では、会話が自動的に保存されます。左側のサイドバーから過去の会話にいつでもアクセスできます。

- **会話の名前変更:** サイドバーの会話名にマウスを合わせ、メニュー (...) から「Rename」を選択します
- **会話の削除:** 同じメニューから「Delete」を選択します
- **新しい会話の開始:** 画面左上の「New chat」ボタン、またはサイドバー上部のボタンをクリックします

12.3 Artifacts — 会話の横で「モノ」を作る

Artifacts（アーティファクト）は、Claude が生成したコンテンツを会話の右側パネルに表示する機能です。文章、コード、図表、さらにはインタラクティブ（操作できるアプリケーション——ボタンを押したり入力したりできるもの）なアプリケーションまで、さまざまなものを会話しながら作成・編集できます。

12.3.1 Artifacts の活用例

1. ドキュメントの作成

「企画書のドラフトを作って」と依頼すると、右側のパネルに整形されたドキュメントが表示されます。「もっとカジュアルなトーンにして」「第3章を詳しくして」など、会話を続けながら内容を修正できます。完成したら、コピーしたりダウンロードしたりできます。Word ファイルや PDF としてダウンロードすることも可能です。

2. コードの生成と実行

プログラミングの知識がなくても、「BMI を計算するツールを作って」のように依頼すると、Claude が動作するコードを生成し、Artifact 上でそのまま実行できます。HTML、CSS、JavaScript で構成されたウェブアプリケーションなら、その場でプレビューを確認できます。

3. データの可視化

CSV ファイルを添付して「このデータを棒グラフにして」と依頼すると、Claude がインタラクティブなグラフを Artifact として生成します。グラフの色やレイアウトの変更も、会話で指示するだけです。

4. スライド・プレゼン資料

「〇〇についてのプレゼン資料を作って」と依頼すると、スライド形式の資料を Artifact として作成してくれます。各スライドの内容を会話で調整しながら仕上げられます。

つまづきポイント: Artifacts が表示されない場合は、設定画面で Artifacts 機能が有効になっているか確認してください。また、短い回答（数行程度）では Artifact は生成されず、チャット内にそのまま表示されます。

12.4 Projects — 仕事をまとめて管理する

Projects（プロジェクト）は、関連する会話・ファイル・設定をひとつのワークスペースにまとめる機能です。プロジェクトごとにカスタム指示やナレッジ（参考資料）を設定でき、Claude に毎回同じ前提を説明する手間がなくなります。

12.4.1 Projects の使い方

1. サイドバーの「Projects」セクションから「New Project」をクリックします
2. プロジェクト名と説明を入力します
3. 必要に応じて以下を設定します：

- **カスタム指示:** Claude への共通の指示（例: 「回答は常に日本語で」「専門用語は使わないで」）
- **ナレッジ:** 参考資料としてファイルをアップロード（PDF、Word、テキストなど。1 ファイル最大 30MB）

12.4.2 Projects が役立つ場面

- **ブログ運営:** 過去の記事やスタイルガイドをナレッジに登録し、一貫したトーンで新しい記事を書いてもらう
- **業務マニュアル作成:** 社内資料をアップロードして、マニュアルの作成・更新を依頼する
- **学習・研究:** 論文や教科書をナレッジに登録し、内容について質問したり要約を作成したりする
- **人事:** 面接の質問集と評価基準をナレッジに登録し、職種ごとの面接準備を効率化する
- **法務:** 契約書テンプレートをアップロードして、新しい取引先との契約書レビューや修正案の作成を依頼する

Free プランでの制限: Free プランではプロジェクトを最大 5 つまで作成できます。有料プランでは無制限です。また、有料プランではナレッジの検索性能が強化されており、大量の資料を登録しても必要な情報を正確に見つけて回答に反映してくれます。

12.5 Deep Research —— 複雑な調査を Claude に任せる

Deep Research（ディープリサーチ）は、Claude がウェブ上の情報を自律的に調査し、包括的なレポートを作成してくれる機能です。通常の Web 検索が「素早い回答」を返すのに対し、Deep Research は数分〜数十分かけて複数の情報源を調べ、引用付きの詳細なレポートを生成します。

12.5.1 Deep Research の使い方

1. チャット画面の下部にある「Research」ボタンをクリックして有効にします（ボタンが青色になります）
2. 調べたいテーマを入力して送信します
3. Claude が調査計画を立て、必要に応じて確認の質問をしてくれます
4. 調査が始まると、Claude は複数の情報源を横断的に検索・分析します
5. 5〜30 分程度で、引用付きの詳細なレポートが生成されます

12.5.2 Deep Research が向いているテーマ

- **市場調査:** 「2026 年の日本の SaaS 市場の動向をまとめて」
- **競合分析:** 「主要なプロジェクト管理ツールの機能と価格を比較して」
- **業界トレンド:** 「2026 年の業界トレンドレポートを作って」
- **制度調査:** 「各社のリモートワーク制度を比較して、導入のポイントをまとめて」
- **人事・採用:** 「エンジニア採用市場の最新動向と、効果的な求人票の書き方を調べて」

つまづきポイント: Deep Research は月あたりの利用回数に制限があります。簡単な質問には通常の会話を使い、本格的な調査が必要ときだけ Deep Research を活用しましょう。

12.6 Memory — Claude があなたを覚えてくれる

Memory（メモリー）は、Claude がユーザーの好みや情報を会話をまたいで記憶する機能です。2026 年 3 月時点で、Free プランを含むすべてのプランで利用できます。

12.6.1 Memory でできること

- **名前や職業の記憶:** 「私は田中です、マーケティング部で働いています」と伝えと、以降の会話で自動的に考慮されます
- **好みの記憶:** 「回答はいつも箇条書きでお願い」「カジュアルな文体が好き」といった好みを覚えてくれます
- **進行中の作業の記憶:** 「今、新商品の企画書を作っています」と伝えれば、次の会話でも続きから進められます

12.6.2 Memory の管理

- Claude に「これを覚えて」と伝えることで、明示的に情報を記憶させることができます
- 設定画面からメモリーの内容を確認・編集・削除できます
- プライベートな会話をしたい場合は、インコグニートチャット（匿名チャット）を使えば、メモリーに保存されません。チャット画面で新しい会話を開始する際に「Incognito」オプションを選択してください。なお、これはブラウザのシークレットモード（プライベートブラウジング）とは別の機能です

つまづきポイント: Memory はプロジェクトごとに分離されています。仕事のプロジェクトで覚えた内容は、プライベートのプロジェクトには影響しません。ただし、Projects に紐づかない通常の会話での Memory はすべての会話に共通で適用されます。

12.7 Connectors — 外部サービスと連携する

Connectors（コネクターズ）は、Claude Web に外部サービスを接続する機能です。150 以上のコネクタが用意されており、コーディング、データ分析、デザイン、金融、営業など幅広い分野のサービスと連携できます。Free プランを含むすべてのプランで利用可能です。

12.7.1 Connectors でできること

コネクタを追加すると、Claude が外部サービスの情報を直接参照できるようになります。

- **Microsoft 365 連携:** Outlook のメール、OneDrive のドキュメント、SharePoint のファイルを Claude の会話に持ち込めます。「今日届いたメールを要約して」「OneDrive の報告書を分析して」といった使い方が可能です
- **開発ツール連携:** GitHub、GitLab などのリポジトリやイシューを参照できます
- **ビジネスツール連携:** Slack、Notion、Google Drive など、日常業務で使うサービスとつながられます

12.7.2 コネクタの設定方法

1. claude.ai/customize/connectors にアクセスします
2. 利用したいコネクタを選択します
3. 対象サービスのアカウントでログイン（認証）します
4. 接続が完了すると、会話の中でそのサービスの情報を参照できるようになります

12.7.3 自作コネクタの提出

公式ディレクトリにないサービスとの連携が必要な場合、自作コネクタを提出して審査を受けることもできます。コネクタは MCP（Model Context Protocol）をベースに構築されており、技術的な詳細は「[MCP ガイド](#)（ページ 345）」を参照してください。

つまづきポイント: 組織の Microsoft 365 アカウントで接続する場合、IT 管理者がサードパーティアプリの接続を制限していることがあります。接続できない場合は社内の IT 担当者に確認してください。

12.8 モデルの切り替え —— 用途に合わせて選ぶ

Claude Web では、会話ごとに使用するモデル（AI の「頭脳」にあたるもの）を切り替えることができます。モデルによって、回答のスピードや精度が異なります。

12.8.1 モデルの種類と特徴

モデル	特徴	向いている用途
Haiku	最も高速で軽量	簡単な質問、短い文章の生成、ちょっとした翻訳
Sonnet	速度と性能のバランスが良い（標準モデル）	日常的な作業全般。文章作成、分析、コード生成
Opus	最も高性能で深い思考が可能	複雑な分析、長文の構成、難しい問題の解決

12.8.2 モデルの選び方ガイド

- **迷ったら Sonnet:** 日常の 80% の作業は Sonnet で十分です。速度と品質のバランスが最も優れています
- **急いでいるなら Haiku:** 素早い応答が欲しいときや、シンプルな作業には Haiku が最適です
- **じっくり考えてほしいなら Opus:** 重要な文章の作成、複雑なデータ分析、深い考察が必要なときに選びましょう

Free プランでの制限: Free プランでは Sonnet と Haiku を利用できます。Opus は Pro 以上のプランで利用可能です。

チャット画面の上部にあるモデル名をクリックすると、ドロップダウンメニューからモデルを切り替えら

れます。会話の途中でも切り替え可能です。

12.9 次のステップ

Claude Web の基本的な使い方を理解できたら、さらにステップアップしましょう。

- **Claude とは？** (ページ 3) --- Claude の全体像を把握する
- **料金プランの比較** (ページ 15) --- 自分に合ったプランを選ぶ
- **Claude デスクトップアプリの使い方** (ページ 171) --- PC にインストールして使う Claude を試す
- **プロンプトの基本** (ページ 239) --- Claude にうまく指示を出すテクニックを学ぶ

第13章 Artifacts — 会話の横で「モノ」を作る

Artifacts（アーティファクト）は、Claude との会話の中で生まれたコンテンツを、チャット画面の右側に独立したパネルで表示する機能です。文章やコード、グラフ、さらには操作できるミニアプリまで、幅広いものを作りながら会話を続けられます。

Free プランを含むすべてのプランで利用できます。 プログラミングの知識は一切不要で、普通の言葉で依頼するだけで動作するアプリケーションも作れます。

この記事では、Artifacts の基本的な仕組みから具体的な業務活用まで、丁寧に解説します。

13.1 Artifacts とは

通常の Claude との会話では、回答はチャットの流れの中にそのまま表示されます。一方、Artifacts が有効になると、ある程度まとまった分量・形式を持つコンテンツは、チャット画面とは別の専用パネル（右側に開く）に表示されます。

13.1.1 通常の回答と何が違う？

	通常の回答	Artifacts
表示場所	チャットの流れの中	右側の独立パネル
編集のしやすさ	スクロールして探す必要がある	パネルに常時表示されている
コピー・保存	テキストを手動で選択	ボタン1つでコピー・ダウンロード
バージョン管理	過去の版は残らない	修正のたびに版が記録される

13.1.2 いつ Artifacts が生成されるか

Claude は以下の条件を満たすコンテンツを自動的に Artifact として生成します。

- 内容がある程度まとまっている（おおむね 15 行以上）
- 後から参照・修正する可能性が高い
- 会話の文脈なしに単独で成り立つ

逆に、短い回答や一言の返答は通常どおりチャット内に表示されます。

13.1.3 Artifacts の有効化方法

Artifacts が表示されない場合は、設定を確認してください。

1. 画面左下のアカウント名（またはイニシャル）をクリックします
2. 「Settings（設定）」を開きます
3. 「Capabilities（機能）」タブを選択します
4. 「Artifacts」のトグルをオンにします

つまづきポイント: 設定を有効にしても Artifacts が表示されない場合は、ブラウザのページを更新 (F5 キー) してみてください。また、依頼内容が短すぎると Artifact は生成されず、チャット内に表示されます。「○○について詳しくまとめて」のように、まとまった量のアウトプットを求める指示を入れると確実です。

13.2 Artifacts の種類

Artifacts で生成できるコンテンツには、大きく分けて以下の種類があります。

13.2.1 ドキュメント（Markdown 形式）

報告書、企画書、議事録、手順書など、文字ベースのドキュメントです。見出しや箇条書き、表などが整形された状態で表示されます。

例: 「新製品の発売に向けた販売計画書を作って」「先週のミーティングの議事録を要約して」

13.2.2 コード

Python、JavaScript、HTML、CSS など各種プログラミング言語のコードです。エンジニア以外の方は直接使用の機会は少ないかもしれませんが、後述のアプリ作成に内部的に使われています。

13.2.3 HTML アプリ（インタラクティブ）

ブラウザ上で動作する Web ページやミニアプリです。ボタンをクリックしたり、数値を入力したりして操作できます。プログラミングの知識がなくても、「○○を計算するツールを作って」と依頼するだけで動くアプリが出来上がります。

例: 計算ツール、チェックリスト、タイマー、簡単なゲーム

13.2.4 SVG 画像・図解

SVG（スケーラブルベクターグラフィックス — 拡大縮小しても画質が劣化しない形式のデータ画像）として作成されたイラストや図解です。フローチャート、組織図、シンプルなアイコンなどを作れます。

例: 業務フローの図解、組織図、インフォグラフィック

13.2.5 Mermaid 図表

Mermaid（マーメイド — テキストをもとに図表を自動生成するツール）を使ったフローチャートや順序図、ガントチャートなどです。

例: プロジェクトの工程管理チャート、システムの流れ図

13.3 Artifacts の基本操作

13.3.1 生成する

特別な操作は必要ありません。Claude に依頼するだけで、内容に応じて自動的に Artifact が生成されます。

企画書のドラフトを書いてください。

テーマは「社内の業務効率化 DX 推進」で、

背景・目的・施策案・スケジュール・予算感の構成でまとめてください。

13.3.2 会話で修正する

Artifact が表示された後も、会話を続けることで内容を修正できます。チャット欄に修正指示を入力するだけです。

修正指示の例

「第 2 章をもっと具体的にしてください」

「全体をもっとシンプルな言葉に書き直して」

「予算感のセクションに別案も追記してください」

修正するたびに Artifact の内容が更新されます。

13.3.3 バージョンを切り替える

修正を重ねると、Artifact パネルの上部にナビゲーション（← →）が表示されます。これをクリックすることで、過去のバージョンに戻ったり、比較したりできます。

「一つ前の版の方がよかった」と感じたときに便利です。

13.3.4 コピー・ダウンロードする

Artifact パネルの右下にあるボタンから、コンテンツをコピーしたりダウンロードしたりできます。

- **コピー**: クリップボードにテキストをコピーします
- **ダウンロード**: ファイルとしてダウンロードします。Word（.docx）、PDF、Excel（.xlsx）、PowerPoint（.pptx）など、内容に応じた形式で保存できます

13.3.5 公開・共有する

完成した Artifact は、リンクを作成して他の人と共有できます。Claude アカウントを持っていない人でもリンクを開いて閲覧・操作できるため、社外の方への共有にも使えます。

Artifact パネルの上部にある「Share」ボタンをクリックすると、公開リンクが生成されます。また、Web サイトへの埋め込み用コードも取得できます。

つまづきポイント: 一度「非公開」に設定した Artifact は、同じ Artifact を再び公開することができません。再度公開したい場合は、新しく Artifact を作り直す必要があります。大切な Artifact のリンクは消す前に手元に保存しておきましょう。

13.4 活用例：ビジネス文書の作成

Artifacts は企画書、報告書、議事録などのビジネス文書作成に威力を発揮します。

13.4.1 企画書の作成

以下のようなプロンプトをそのままコピーして使えます。

以下の情報をもとに、社内提案用の企画書を作成してください。

テーマ：カスタマーサポートへの AI チャットボット導入

背景：問い合わせ件数が月 500 件を超え、対応が追いついていない

目的：一次対応の自動化で、担当者の工数を 30% 削減する

対象読者：経営企画部・情報システム部

構成：背景と課題 / 提案内容 / 期待効果 / スケジュール / コスト概算

13.4.2 議事録の整理

会議のメモや音声文字起こしを貼り付けて、整形された議事録を作ってもらえます。

以下は本日の会議の録音を文字起こしした内容です。

これをもとに、以下の構成で議事録を作成してください。

- ・日時・参加者・議題
- ・各議題の議論内容（箇条書き）
- ・決定事項
- ・次回アクション（担当者・期限付き）

【文字起こし内容】

（ここに文字起こしを貼り付ける）

13.4.3 操作マニュアルの作成

社内ツールや業務手順のマニュアルも Artifact で作れます。

社内の経費精算システム（〇〇）の操作マニュアルを作成してください。
対象は入社3年目以内の一般社員で、
PC 操作に不慣れな人でもわかるよう、ステップバイステップで丁寧に説明してください。
主な操作：領収書の登録 / 申請書の作成・提出 / 承認状況の確認

つまづきポイント: 指示が曖昧だと、Claude の解釈で内容が決まってしまう。「対象読者は誰か」「どんな構成にしたいか」「どのくらいの長さか」を最初から明示すると、修正が少なく済みます。

13.5 活用例：データの可視化

数字やデータを視覚的に表現するグラフやダッシュボードも、Artifacts で簡単に作れます。プログラミングは一切不要です。

13.5.1 CSV データからグラフを作る

月次売上データなどの CSV ファイルを添付してから、以下のように依頼します。

添付の CSV ファイルは、2025 年 1 月～12 月の月次売上データです。
以下のグラフを作成してください。

- ・月別売上の棒グラフ（前年比も重ねて表示）
- ・商品カテゴリ別の構成比を円グラフで
- ・月ごとに色分けして、全体のトレンドがわかるようにしてください

13.5.2 数値データを直接入力して可視化する

CSV がなくても、数字を直接貼り付けてグラフを作れます。

以下のデータを使って、月別のアクティブユーザー数の推移を
折れ線グラフで可視化してください。
グラフには目標ライン（5,000 人）も追加してください。

1 月：3,200 人
2 月：3,800 人
3 月：4,100 人
4 月：4,600 人
5 月：4,900 人
6 月：5,300 人

13.5.3 KPI ダッシュボードを作る

複数の指標をまとめて表示するダッシュボードも作れます。

以下の KPI を一覧表示するシンプルなダッシュボードを HTML で作ってください。
それぞれ「今月の実績」と「達成率」を表示し、
達成率が 80% 以下の場合は赤色、80～100% は黄色、100% 以上は緑色で表示してください。

売上：8,500 万円（目標：1 億円）
新規顧客数：42 社（目標：50 社）
継続率：94%（目標：95%）
NPS スコア：52（目標：50）

つまづきポイント: グラフのデザインや色が気に入らない場合は、「青とオレンジの 2 色だけに
して」「背景を白にして」「フォントを大きくして」のように、会話で具体的に指示すれば修正
できます。

13.6 活用例：ミニアプリ・ツールの作成

Artifacts の最も驚かれる使い方が、インタラクティブなアプリの作成です。HTML と JavaScript で動くウェブアプリを、プログラミングの知識なしに作れます。

13.6.1 計算ツールを作る

業務でよく使う計算をアプリ化できます。

以下の機能を持つ見積金額計算ツールを HTML で作成してください。

- ・サービスの種類（A プラン / B プラン / C プラン）をドロップダウンで選択
- ・数量を入力するフィールド
- ・割引率（0～30%）をスライダーで調整
- ・税込み・税抜き金額をリアルタイムで計算して表示
- ・「見積書を印刷」ボタンを付ける

デザインはシンプルでビジネスっぽく、スマホでも使いやすい画面にしてください。

13.6.2 チェックリストやフォームを作る

社内手順の確認チェックリストや、簡易アンケートフォームも作れます。

新入社員オンボーディングのチェックリストアプリを作ってください。

- ・入社前 / 入社初日 / 入社 1 週間 / 入社 1 ヶ月のタブで切り替えられる
- ・各タスクにチェックボックスがあり、チェックするとグレイアウトされる
- ・上部に全体の進捗バー（%）を表示
- ・完了したタスクの数と残りのタスク数も表示する

13.6.3 クイズ・学習ツールを作る

社内研修用のクイズや、製品知識の確認ツールも簡単に作れます。

社内セキュリティ研修用の確認クイズアプリを作ってください。

問題は 5 問で、各問題に 4 択の選択肢を表示します。

回答後に正解・不正解を表示し、解説も表示してください。

最後に合計スコアを表示して、80 点以上なら「合格」と表示してください。

問題内容：

1. パスワードの最低文字数は？ → 正解：12 文字以上
2. フィッシングメールの見分け方は？ → 正解：差出人のドメインを確認
(以降、問題を追加)

つまづきポイント: 作ったアプリは Artifact パネル内でそのまま操作できます。「動作確認してみたい」と思ったら、パネル内のボタンやフォームを直接クリック・入力してみてください。思った通りに動かない部分があれば「○○の部分動かないので直して」と伝えるだけで修正できます。

13.7 Artifacts カタログ — 他の人が作ったものを活用する

claude.ai/catalog/artifacts では、世界中のユーザーが作って公開した Artifacts を無料で閲覧・利用できます。

13.7.1 カタログでできること

- **ブラウズ:** ツール、教育、ゲーム、ビジュアライゼーションなど、カテゴリ別に公開 Artifacts を探せます
- **即実行:** 気になる Artifacts はリンクを開くだけですぐに使えます。Claude アカウントがなくても閲覧・操作できます
- **カスタマイズ:** ログインした状態でカタログの Artifacts を開くと、自分用にアレンジするための会話を Claude と始められます

13.7.2 自分の Artifacts を公開する

気に入った Artifacts は、カタログに公開してコミュニティと共有できます。

1. Artifact パネルの上部にある「Share」ボタンをクリックします
2. 「Publish to catalog」オプションを選択します
3. タイトルと説明を入力して公開します

つまづきポイント: カタログで見つけた Artifacts を「自分の業務用にアレンジしたい」場合は、その Artifacts を開いた状態で Claude に依頼するのが最短ルートです。「この計算ツールに消費税の設定項目を追加して」のように話しかければ、元のコードをベースに修正してくれます。

13.8 次のステップ

Artifacts を使いこなせるようになったら、Claude Web の他の機能も活用してみましょう。

- *Claude Web* (*claude.ai*) の使い方 (ページ 89) — Artifacts 以外の機能 (Projects、Deep Research、Memory など) をまとめて確認する
- *Deep Research* — 複雑な調査を *Claude* に任せる (ページ 105) — 作成した文書の根拠となる調査を *Claude* に依頼する
- *Memory* — *Claude* があなたを覚える (ページ 113) — よく使う Artifacts のスタイルを記憶させて、毎回の指示を省く

第14章 Deep Research — 複雑な調査を Claude に任せる

「市場調査をしたいけど、どこから調べればいいかわからない」「競合他社の情報をまとめるのに丸一日かかってしまう」——そんな悩みを持つビジネスパーソンのために、Claude には **Deep Research**（ディープリサーチ）という機能があります。

Deep Research を使うと、Claude が自動的にウェブ上の多数の情報源を横断して調査し、引用付きの詳細なレポートを作成してくれます。この記事では、その仕組みと使い方を丁寧に解説します。

14.1 Deep Research とは

Deep Research は、通常の AI チャットとは根本的に異なるアプローチで情報を調べる機能です。

14.1.1 通常の会話との違い

普段 Claude に質問すると、Claude は自分の知識をもとに回答します。Web 検索を使う場合でも、1〜2 回の検索で情報を取得するのが一般的です。

Deep Research は違います。**複数の AI（マルチエージェント）が役割分担して、並行して大量の情報源を調べ上げる**仕組みになっています。

具体的には次のような流れで動いています。

1. **リードエージェント**（指揮役）がリサーチの戦略を立てる
2. **複数のサブエージェント**（実行役）が異なる角度から同時に検索・分析する
3. リードエージェントが収集した情報を統合し、一貫したレポートにまとめる

この仕組みにより、人間が手作業で行うような「関連サイトを数十件読んで、重要な情報を整理する」という作業を、Claude が自動でこなしてくれます。

14.1.2 Deep Research でできること

- 複数のウェブサイト・レポート・ニュース記事を横断した調査
- 出所（引用元 URL）付きのレポート生成
- 情報の比較・対照（競合比較、制度比較など）
- 最新の情報を含む包括的なまとめ

注意: Deep Research が生成したレポートの情報は、必ずしもすべて正確とは限りません。AI が誤った情報を自信ありげに述べてしまう「ハルシネーション」が起こる可能性があります。重要な意思決定に使う場合は、必ず引用元の原文を確認してください。

14.2 使い方

Deep Research の操作は難しくありません。「Research」ボタンを有効にして、調べたいテーマを入力するだけです。

14.2.1 ステップ 1: Research ボタンを有効にする

チャット画面の入力欄近くにある「Research」ボタンをクリックします。ボタンが青色になれば有効な状態です。

つまづきポイント: 「Research」ボタンが見当たらない場合は、有料プランにアップグレードしているか確認してください。Deep Research は Pro・Max・Team・Enterprise プランのみで利用できます。Free プランでは使用できません。

14.2.2 ステップ 2: 調査テーマを入力する

Research が有効な状態で、調べたいテーマをチャット欄に入力して送信します。

2026 年の日本の HR テクノロジー市場の動向と、主要なプレイヤーを調べてまとめてください。中小企業向けの採用管理ツールを中心に、機能比較と価格帯もわかるようにお願いします。

14.2.3 ステップ 3: 調査計画を確認する

Claude はまず**調査計画**を提示します。「どんな観点で調べるか」を事前に確認してくれるので、方向性が合っているか確認しましょう。

もし方向性が違う場合は、「競合比較よりも価格動向を詳しく調べてほしい」などと返信して修正できます。問題なければ進めるよう伝えましょう。

また、Claude が「具体的にどの業界を対象にしますか？」のような確認質問をしてくることもあります。できるだけ具体的に答えると、より精度の高いレポートが生成されます。

14.2.4 ステップ 4: レポートを受け取る

調査が始まると、Claude は自動的に複数の検索を実行し始めます。画面には現在調べている内容が随時表示されます。

所要時間の目安は次のとおりです。

テーマの複雑さ	目安時間
シンプルな調査（競合 3〜5 社の比較など）	5〜10 分
中程度の調査（市場動向 + 競合分析）	10〜20 分

次のページに続く

表 1 – 前のページからの続き

テーマの複雑さ	目安時間
複雑な調査（業界全体のトレンド + 規制 + 事例）	20～30 分

調査中はそのままページを開いたままにしておきましょう。タブを閉じると調査が中断される場合があります。

14.3 向いているテーマ・向いていないテーマ

Deep Research は強力な機能ですが、すべての質問に向いているわけではありません。用途に合わせて使い分けることが大切です。

14.3.1 向いているテーマ

市場調査・業界分析

- ・「2026 年の国内 EC 市場の規模と成長トレンド」
- ・「食品業界における DX 推進の現状と課題」

競合分析

- ・「主要なプロジェクト管理ツール（Asana、Notion、Monday.com 等）の機能・価格比較」
- ・「国内の採用管理システムの比較と各社の強み」

業界トレンド・技術動向

- ・「生成 AI を活用したカスタマーサポートの最新事例」
- ・「ESG 経営に取り組む日本企業の事例集」

制度・規制の調査

- ・「改正育児・介護休業法の概要と企業の対応状況」
- ・「個人情報保護法改正が SaaS 企業に与える影響」

人事・採用

- ・「エンジニア採用市場の最新動向と効果的な求人票の書き方」
- ・「リモートワーク制度の先進事例と導入ポイント」

14.3.2 向いていないテーマ

テーマ	理由	代わりにすること
「今日の株価を教えて」などリアルタイム性が必要な情報	Web 情報に時間差があるため	専用の金融サービスを使う

次のページに続く

表 2 – 前のページからの続き

テーマ	理由	代わりにすること
「〇〇さんの連絡先を調べて」など個人情報収集	プライバシーの問題	公式の連絡先を直接確認
「うちの会社の売上傾向は？」など内部データが必要な質問	社外の情報しか調べられないため	社内データを添付して通常チャットで分析
ごく限られた分野や超ニッチな専門知識	Web 上の情報が少ないため精度が落ちる	専門家や文献を直接参照

つまづきポイント: 「なんとなく調べてほしい」という曖昧なテーマは避けましょう。Deep Research はテーマが明確なほど精度の高いレポートを生成します。曖昧なテーマで調査を始めると、方向性が外れたレポートが生成されることがあります。

14.4 良い調査依頼の書き方

Deep Research の品質は、依頼の書き方で大きく変わります。

14.4.1 具体的な依頼 vs 曖昧な依頼

曖昧な依頼 (NG 例)

マーケティングについて調べて

この依頼では、Claude は「何について」「どの観点で」「どの程度詳しく」調べればいいのかわかりません。

具体的な依頼 (OK 例)

2026 年の日本における B2B 向けコンテンツマーケティングの最新トレンドを調査してください。

調査してほしい内容：

1. 効果的なコンテンツ形式（ホワイトペーパー、動画、ウェビナー等）とその活用事例
2. 主要なプラットフォーム（LinkedIn、note 等）ごとの特徴と使い分け
3. 生成 AI がコンテンツ制作に与えている影響と先進企業の対応事例

対象：従業員 100～1000 名規模の IT 系企業のマーケティング担当者向けの視点で

14.4.2 業務シナリオ別のプロンプト例

人事・採用担当者向け

2026 年の IT エンジニア採用市場の現状を調査してください。

以下の点を含めてまとめてください：

- 求人倍率の推移と採用難易度の変化
- エンジニアが転職先を選ぶ際の重視ポイント（給与・リモートワーク・技術スタック等）

(次のページに続く)

(前のページからの続き)

- 中小企業が大企業と差別化して採用を成功させている事例 3～5 件
- 効果的な求人票の書き方と記載すべき情報

マーケティング担当者向け

競合他社 3 社 (A 社、B 社、C 社) のデジタルマーケティング戦略を比較調査してください。

比較してほしい項目：

- 各社のコンテンツ戦略 (SNS、ブログ、メルマガ等)
- SEO・検索流入の状況 (推定でよい)
- 広告出稿の傾向 (Google 広告・SNS 広告等)
- ターゲットとしている顧客層

自社の戦略改善に活かせる示唆も加えてください。

法務・コンプライアンス担当者向け

2025 年～2026 年に施行・改正された、SaaS 企業が遵守すべき主な法律・規制を調査してください。

対象：国内向けに B2B SaaS を提供する企業

含めてほしい情報：

- 法律・規制の名称と施行日
- 企業に求められる具体的な対応内容
- 違反した場合のペナルティ
- 先行して対応した企業の事例 (あれば)

14.4.3 Claude が確認質問をしてきたら

Deep Research を開始すると、Claude が調査方針の確認や追加情報の質問をすることがあります。たとえば次のようなメッセージです。

「調査対象の地域は日本国内のみですか？それとも海外も含めますか？」

こうした質問には、できるだけ具体的に答えましょう。回答が曖昧だと、後でレポートの方向性がずれてしまうことがあります。急いでいる場合は「日本国内のみ、2026 年時点の情報を優先してください」のようにポイントを絞って答えるのが効果的です。

14.5 レポートの読み方と活用

14.5.1 引用の確認方法

Deep Research のレポートには、各情報の出所として引用元の URL が含まれています。レポート内の番号 ([1][2] のような表記) をクリックすると、元の情報源にアクセスできます。

重要な情報は、必ず引用元の原文を確認する習慣をつけましょう。Claude が「A という調査によると売上は 30% 増加」と書いていても、実際の資料には「一部の企業において 30% 増加」と書かれている場合があります。

14.5.2 レポートの信頼性を見極めるポイント

チェック項目	確認すること
引用元の質	公式サイト・学術論文・大手メディアなど信頼性の高いソースか
情報の新しさ	引用元の日付が古くないか（特に変化の速い業界では重要）
情報の一致	複数の引用元が同じ内容を指示しているか
推測と事実の区別	「～と考えられる」「～の可能性がある」という表現に注意

14.5.3 社内資料への転用

Deep Research のレポートは、そのまま社内資料のたたき台として使えます。活用例をいくつか紹介します。

企画書・提案書のバックデータとして 「この提案の根拠となる市場データ」として、レポートの要点を引用できます。その際は引用元の URL を明記しましょう。

社内勉強会の資料として 業界トレンドや技術動向のレポートは、社内共有用のサマリー資料として加工しやすいです。

議事録・会議資料の参考資料として 特定のテーマについて議論する前に Deep Research でバックグラウンドを調査しておく、会議の質が上がります。

Artifact でレポートを整形する

レポートを受け取ったら、続けて次のように依頼してみましょう。

このレポートの内容を、役員向けの報告書として A4 1 ページにまとめた Artifact を作成してください。

Claude が、レポートの内容をベースにした整形済みドキュメントを Artifact として生成してくれます。そのまま Word でダウンロードして利用できます。

つまづきポイント: レポートをそのままコピーして社外に提出するのはリスクがあります。引用元の確認なしに情報を使うと、誤情報を広めてしまう可能性があります。必ず一次情報を確認してから使用しましょう。

14.6 利用上限と料金

14.6.1 対応プランと制限

Deep Research は、**有料プランのみ**で利用できます。Free プランでは使用できません。

注意: 料金は変更される場合があります。最新の情報は [Anthropic 公式の料金ページ](#) で確認してください。

プラン	月額料金	Deep Research の利用
Free	無料	利用不可
Pro	\$20	利用可能（利用制限あり）
Max 5x	\$100	Pro の 5 倍の利用量
Max 20x	\$200	Pro の 20 倍の利用量
Team / Enterprise	要確認	利用可能

14.6.2 トークン消費について

「トークン」とは、AI が処理するテキストの単位です（1,000 トークン ≒ 日本語 500～700 文字程度）。プランの利用制限はトークン消費量で計算されます。

Deep Research は通常の会話より大幅にトークンを消費します。

利用方法	トークン消費の目安
通常のチャット	1x（基準）
Web 検索付きチャット	約 4 倍
Deep Research	約 15 倍

Deep Research 1 回のリサーチで、通常の会話 15 回分程度のトークンを消費するイメージです。

14.6.3 効率的に使うコツ

- **簡単な質問は通常のチャットで:** 「○○について教えて」レベルの質問には Deep Research は使わず、通常会話を使いましょう
- **週に数回の本格調査に絞る:** 月のトークン上限を考慮し、本当に深く調べたいテーマを厳選して使うのが効果的です
- **1 回の依頼でまとめる:** 関連するテーマはひとつのリサーチ依頼にまとめると、トークンを節約できます

つまづきポイント: 「利用制限に達しました」というメッセージが出た場合は、利用制限がリセットされるまで待つか、プランのアップグレードを検討してください。利用制限は 5 時間のローリングウィンドウ方式でリセットされます（すべての有料プラン共通）。

14.7 次のステップ

Deep Research の使い方を理解したら、さらにステップアップしましょう。

- [Claude Web \(claude.ai\) の使い方](#) (ページ 89) — Deep Research を含む、Claude Web 全体の機能を確認する
- [プロンプトの基本](#) (ページ 239) — より精度の高い調査依頼を書くテクニックを学ぶ
- [料金プランの比較](#) (ページ 15) — 自分の利用頻度に合ったプランを選ぶ

第15章 Memory — Claude があなたを覚えてくれる

人間の同僚と長く一緒に働くと、相手はあなたの仕事のスタイルや好みを自然に把握してくれます。Claude の **Memory**（メモリー）機能は、これと同じようなことを AI との会話でも実現します。

Memory を使うと、Claude があなたの名前、職業、回答スタイルの好み、進行中のプロジェクト情報などを記憶してくれます。次の会話からは「毎回同じ前提を説明する手間」がなくなり、すぐに本題に入れるようになります。

2026 年 3 月から、Memory 機能は **Free プランを含む全プラン**で利用できるようになりました。

15.1 Memory とは

Memory は、Claude があなたに関する情報を会話をまたいで保持する機能です。

通常の AI チャットでは、会話を始めるたびに Claude はあなたのことを何も知らない状態からスタートします。毎回「私は〇〇会社のマーケティング担当で、日本語で箇条書きの回答を好みます」と説明し直すのは手間がかかります。

Memory を有効にすると、Claude は会話の内容を分析して重要な情報を自動的に要約・記録します。次の会話では、その記録を参照しながら回答してくれます。

15.1.1 Memory の仕組み

Memory は 24 時間ごとに更新されます。Claude が過去の会話から「仕事に関連する重要な情報」を抽出して要約し、保存します。この仕組みを「Memory Synthesis（メモリーシンセシス）」と呼びます。

要約された情報は、新しい会話を始めるたびに自動的に Claude に読み込まれます。

補足: Memory は「全会話をそのまま保存して検索する」RAG（検索拡張生成）ではなく、「重要な情報を要約して記録する」抽出型の仕組みです。過去の会話を丸ごと参照したい場合は、後述の Chat Search（有料プラン限定）を使います。

15.2 Memory に覚えてもらえること

Claude が記憶できる情報はさまざまです。仕事の効率化に役立つ具体的な例を紹介します。

15.2.1 あなた自身の情報

- **名前・役職:** 「私は田中花子です、人事部の採用担当をしています」
- **業種・会社:** 「IT スタートアップで働いています、従業員は 50 人ほどです」
- **スキルレベル:** 「Excel は初心者レベルです、プログラミングの経験はありません」
- **使用言語:** 「回答は常に日本語をお願いします」

15.2.2 回答スタイルの好み

- **フォーマット:** 「回答はいつも箇条書きにしてください」
- **長さ:** 「要点を 3 つ以内にまとめて」
- **文体:** 「丁寧語より普通の話し言葉で話してほしい」
- **専門用語:** 「IT 用語は使わず、わかりやすい言葉で説明してください」

15.2.3 進行中のプロジェクト情報

- 「現在、新商品のマーケティング戦略を立案中です」
- 「来月に向けて採用面接の準備をしています」
- 「社内の DX 推進プロジェクトのリーダーをしています」

15.2.4 よく使うフォーマット・テンプレート

- 「議事録は『決定事項・アクションアイテム・次回予定』の順でまとめてください」
- 「メールの件名は常に『【要確認】』から始めてください」

つまづきポイント: Memory は「仕事に関連する情報」を優先的に記憶します。趣味や日常的な雑談は記録されにくいことがあります。確実に覚えてほしい情報は、次のセクションで紹介する「明示的に覚えてもらう方法」を使いましょう。

15.3 Memory の使い方

15.3.1 自動記憶（会話の中で自然に覚えていく）

Memory を有効にしていれば、特別な操作は必要ありません。Claude との会話の中で伝えた情報が、24 時間ごとに自動的に整理・保存されます。

例えば、今日の会話で「私は営業担当の山田です」と話したとすると、翌日以降の会話では Claude があなたを「営業担当の山田さん」として扱ってくれます。

15.3.2 明示的に覚えてもらう

「これを確実に覚えてほしい」という情報は、直接 Claude に伝えましょう。

指示の例:

「これを覚えておいてください：私は会議の議事録を書くとき、必ず箇条書き形式を使います」

「次の情報を記憶してください：

- 担当：新規事業開発部
- 現在取り組んでいる案件：海外市場への展開戦略
- 苦手なこと：英語でのコミュニケーション」

Claude は「覚えました」と確認してくれます。

15.3.3 設定画面でメモリを管理する

記憶された内容はいつでも確認・編集・削除できます。

1. [claude.ai](#) にログインします
2. 画面左下のアカウントアイコンをクリックし、「Settings」（設定）を開きます
3. 「Capabilities」（機能）のタブを選択します
4. 「Memory」の項目で「**View and edit memory**」をクリックします

表示された画面では以下の操作ができます。

- **内容の確認:** 何が記憶されているかテキストで確認できます
- **編集:** 間違いや古い情報を修正できます
- **一時停止:** 「Pause memory」で、既存のメモリは残したまま新しい記憶を止めます
- **完全削除:** 「Reset memory」で、記録されたすべてのメモリを削除します（取り消しはできません）

つまづきポイント: 「Memory を有効にしたのに、Claude が前の会話のことを覚えていない」と感じる場合があります。これは Memory の更新が 24 時間ごとのためです。今日の会話が反映されるのは、翌日以降になります。すぐに覚えてほしい場合は「明示的に覚えてもらう」方法を使ってください。

15.4 Chat Search（会話履歴の検索）

Chat Search（チャットサーチ）は、過去の会話を検索して内容を参照できる機能です。**Pro・Max・Team・Enterprise など有料プランのみ**で利用できます。

Memory が「重要な情報を要約して記録する」のに対し、Chat Search は「過去の会話を直接検索して参照する」機能です。

15.4.1 Chat Search の使い方

特別な設定は不要です。有料プランに加入していれば、自然な言葉で聞くだけで過去の会話を検索してくれます。

質問の例:

「先月、Q2 のロードマップについて話し合ったことを教えてください」

「以前に私が相談した採用基準の話をまとめてください」

「過去に作った営業メールのテンプレートを探してください」

Claude が「過去の会話を検索しています...」と表示しながら、関連する会話を見つけて回答に反映します。

15.4.2 Chat Search の活用シーン

- **プロジェクトの引き継ぎ:** 「半年前にまとめた課題リストを探して」
- **再利用:** 「以前作ったプレゼンのアウトラインを見つけて」
- **確認:** 「先週 Claude に依頼した文章の原案を見せて」

つまづきポイント: Chat Search は「プロジェクト外のチャット」と「各プロジェクト内の会話（プロジェクトに限定）」を検索対象にします。プロジェクト A の会話は、プロジェクト B からは検索されません。

15.4.3 Chat Search を無効にする

Chat Search を使いたくない場合は、Settings > Capabilities から無効にできます。

15.5 Memory Import（他サービスからの移行）

Memory Import（メモリーインポート）は、ChatGPT や Gemini など蓄積したコンテキストや好みを Claude に移行できる機能です。**全プラン（Free 含む）**で利用できます。

他の AI サービスを使っていた方が Claude に乗り換える際に、一から設定し直す手間を省けます。

15.5.1 Memory Import の手順

ChatGPT からの移行

1. ChatGPT を開き、プロンプトを使って記憶されている情報を書き出します
 - 例: 「今まで記録されている私の好みや情報をすべて出力してください」
2. 出力された内容をコピーします
3. claude.com/import-memory にアクセスします（または Settings > Capabilities > Memory Import）
4. テキストボックスに ChatGPT からコピーした情報を貼り付けます

5. 「インポート」をクリックします
6. 処理が完了するまで最大 24 時間かかります

Gemini・Grok からの移行も同様の手順で行えます。

つまづきポイント: Memory Import は「仕事に関連する情報」を優先的に取り込みます。個人的な情報は完全には移行されない場合があります。また、実験的な機能のため、エラーが発生することもあります。うまくいかない場合は、重要な情報を手動で Claude に伝えましょう。

15.6 プライバシーと安全性

Memory 機能を使う上で、プライバシーについて気になる方も多いでしょう。ここでは、記録された情報の管理方法と、プライバシーを守るための方法を説明します。

15.6.1 記録される情報の範囲

Claude が記憶するのは主に「仕事に関連するコンテキスト」です。

- 職務内容・役職・所属部署
- 進行中のプロジェクト情報
- コミュニケーションスタイルの好み
- 技術的な好み（使用ツールなど）

個人的な雑談や一時的な質問は、通常は記憶されません。

15.6.2 機密情報を Memory に入れないための注意

※ **個人情報・機密情報を Memory に記憶させる場合は注意が必要です。**

- 氏名・住所・電話番号など個人を特定できる情報
- パスワードや認証情報
- 社外秘の業務情報や顧客データ
- 財務情報

これらの情報を Memory に入れると、Claude との会話で意図せず参照される可能性があります。機密性の高い情報は Memory に記録しないようにしましょう。

15.6.3 インコグニートチャット（プライベートな会話）

Memory を使わずに会話したい場合は、インコグニートチャット機能を使いましょう。

インコグニートチャットとは、Memory に情報が記録されない会話モードです。ブラウザのシークレットモードとは異なる、Claude 独自の機能です。

使い方:

1. 新しい会話を開始するとき、チャット入力欄の近くにある「Incognito」オプションを選択します

2. このモードでの会話は Memory に保存されません

機密性の高い業務内容を扱うとき、または Memory をリセットせずに一時的にプライベートな会話をしたときに使えます。

15.6.4 Projects ごとの Memory 分離

Memory はプロジェクトごとに分離されています。これは重要なポイントです。

- 「仕事プロジェクト」で覚えた情報は、「個人プロジェクト」には影響しません
- プロジェクト間で情報が混在することはありません

ただし、Projects に紐づかない通常の会話での Memory は、Projects 外のすべての会話に共通で適用されます。

15.7 効果的な Memory の活用法

Memory 機能を使いこなすことで、日常の業務がより効率的になります。実際の活用パターンを紹介します。

15.7.1 パターン 1: 初回セットアップで一気に覚えてもらう

Memory を有効にしたら、最初に自分のプロフィールをまとめて伝えましょう。

例:

以下の情報を覚えておいてください：

【私について】

- 名前：山田太郎
- 役職：営業部マネージャー（IT 企業勤務）
- チームメンバー：5 名

【コミュニケーションスタイルの希望】

- 回答は箇条書きで、3～5 点にまとめてほしい
- 専門用語は使わず、平易な言葉で
- 回答は 300 文字以内を目安に

【現在のプロジェクト】

- 来期の新規顧客開拓プランを策定中
- 社内の CRM ツール移行プロジェクトをリード中

このような「プロフィールまとめ」を最初に伝えるだけで、以降の会話から効率が大きく上がります。

15.7.2 パターン 2: プロジェクトごとに Memory を活用する

Projects 機能と組み合わせると、用途ごとに最適化されたアシスタントを作れます。

- 「採用プロジェクト」：評価基準・採用ペルソナを記録。面接のたびに説明不要に
- 「マーケティングプロジェクト」：ブランドガイドライン・ターゲット顧客を記録。一貫したトーンで文章を作成
- 「学習プロジェクト」：現在の理解レベル・学習目標を記録。適切な難易度で説明してもらえる

15.7.3 パターン 3: フィードバックで記憶を改善する

「その回答、箇条書きじゃなくて文章でお願い」「やっぱり短めの回答の方がいい」など、会話の中でフィードバックを伝えると、Claude が記憶を更新していきます。

より良い記憶を作るためには、Claude に積極的にフィードバックを伝えましょう。

15.7.4 「毎回説明する手間」がなくなる具体例

Memory なし（毎回の会話でこれを説明する必要がある）：

私は IT 企業の営業マネージャーで、チームは 5 人います。
回答は箇条書きで 3〜5 点にまとめ、専門用語は使わないでください。
現在、来期の新規顧客開拓プランを策定しています。
それを踏まえて、営業戦略のアドバイスをください。

Memory あり（次回からこれだけで OK）：

来期の新規顧客開拓に向けた営業戦略のアドバイスをください。

前提情報は Claude がすでに把握しているので、本題だけ伝えれば十分です。

つまづきポイント：「Memory があるのに、前の話を覚えていないみたい」という場合、最後に会話してから 24 時間以内の情報は、まだ Memory に反映されていない可能性があります。また、設定画面で Memory が「一時停止」になっていないか確認してみてください。

15.8 次のステップ

Memory の基本を理解したら、Claude をさらに使いこなしましょう。

- [Claude Web \(claude.ai\) の使い方](#) (ページ 89) — Memory 以外の機能も含む Claude Web の全体ガイドに戻る
- [Artifacts の使い方](#) (ページ 97) — 会話の中でドキュメントやコードを生成・編集する機能
- [Deep Research の使い方](#) (ページ 105) — 複雑な調査を Claude に任せる機能
- [Projects の使い方](#) (ページ 89) — 会話とナレッジをプロジェクトでまとめて管理する方法

第16章 Projects — 会話とナレッジをまとめて管理する

Claude に仕事の相談をするとき、毎回「私は〇〇会社の営業担当で、回答は箇条書きでお願いします」と説明し直すのは手間がかかります。また、複数の仕事プロジェクトを並行して進めていると、会話がどんどん増えて管理が難しくなることもあります。

Projects（プロジェクト）は、こうした問題を解決するための機能です。関連する会話・ファイル・設定をひとつのワークスペースにまとめられ、プロジェクトごとに Claude への指示や参考資料を設定できます。

この記事では、Projects の基本から応用まで、業務での具体的な活用方法を丁寧に解説します。

16.1 Projects とは

Projects は、Claude との会話を「仕事ごと・テーマごと」に整理できる専用のワークスペースです。

16.1.1 通常の会話との違い

	通常の会話	Projects
会話の整理	一覧に並ぶだけ	プロジェクト単位でまとめられる
Claude への指示	毎回説明が必要	カスタム指示として保存され、自動的に引き継がれる
参考資料の共有	ファイルを毎回添付	ナレッジとして登録しておける
チームでの利用	個人のみ	Team / Enterprise プランでメンバーと共有できる

16.1.2 Projects のうれしいポイント

1. 前提を毎回説明しなくていい

「私は人事担当です」「回答は日本語の箇条書きで」「専門用語は使わないで」—— こうした前提条件をカスタム指示として一度設定しておく、そのプロジェクト内のすべての会話に自動で適用されます。

2. 参考資料をいつでも参照できる

社内マニュアル、製品仕様書、過去の報告書など、業務に必要なファイルをナレッジとして登録しておく、Claude がそれらを参考にして回答してくれます。

3. 会話を整理して管理できる

「採用プロジェクト」「マーケティング施策」「新製品開発」など、テーマ別に会話をまとめられます。後から会話を探すのも楽になります。

16.2 プロジェクトの作り方

Projects は数分で作成できます。手順を確認しましょう。

16.2.1 新規プロジェクトの作成

1. [claude.ai](#) にログインします
2. 左サイドバーの「**Projects**」セクションを探します
3. 「**+ New Project**」または「**New project**」ボタンをクリックします
4. プロジェクト名を入力します（例: 「採用業務」「マーケティング施策 2026」）
5. 「**Create project**」をクリックして完成です

作成直後はカスタム指示とナレッジが空の状態です。次のセクションで設定方法を解説します。

16.2.2 プロジェクト内での会話の始め方

プロジェクトを開いた状態で「**New chat**」ボタンをクリックすると、そのプロジェクトに紐づいた会話が始まります。カスタム指示とナレッジが自動的に読み込まれた状態で Claude と話せます。

16.2.3 会話の管理

プロジェクト内の会話は、以下の操作で管理できます。

- **スター付き**: 重要な会話を目立つようにマーク
- **アーカイブ**: 終了した会話を一時的に非表示にする
- **削除**: 不要な会話を削除する

つまづきポイント: 誤って会話を削除してしまうと復元できません。削除前に内容を確認するか、アーカイブ機能を活用しましょう。

16.3 カスタム指示を設定する

カスタム指示とは、「このプロジェクトの会話では、常にこのように振る舞ってほしい」という Claude への共通の指示です。

16.3.1 カスタム指示の設定方法

1. プロジェクトを開きます
2. 右上または設定アイコンから「**Project settings**」または「**Edit**」を開きます
3. 「**Custom instructions**」（カスタム指示）の欄に指示を入力します
4. 保存します

16.3.2 カスタム指示の具体例

用途別の指示例をそのままコピーして使えます。自分の業務に合わせて内容を変更してください。

営業・提案業務向け

あなたは私の営業アシスタントです。

以下のルールで回答してください：

- 回答は日本語の箇条書き（3～5点）でまとめる
- 具体的な数字や事例を使って説明する
- 専門用語は使わず、顧客にそのまま説明できる言葉を使う
- 提案書・メールを作る場合は、丁寧かつ簡潔なビジネス文体にする

人事・採用業務向け

あなたは人事部の採用担当のアシスタントです。

- 求人票、面接質問、評価コメントなど採用に関する文書を作成するときは、公正で偏りのない表現を使う
- 日本の労働基準法・雇用機会均等法に配慮した内容にする
- 回答は日本語で、社内向けの文書は丁寧語、外部向けは敬語を使い分ける

マーケティング・コンテンツ制作向け

あなたはコンテンツマーケティングのアシスタントです。

ブランドについて：

- ブランド名：〇〇〇
- ターゲット：30～40代の働くビジネスパーソン
- トーン：親しみやすく、実用的。難しい言葉は避ける

記事やSNS投稿を書く際は、上記のブランドガイドラインに沿ってください。

学習・研究向け

私は〇〇を学んでいる初心者です。

説明するときは：

- 中学生でもわかる言葉で、例え話を交えて説明する
- 重要な用語は初登場時に必ず意味を説明する
- 最後に「確認テスト」として簡単なクイズを1問付ける

つまづきポイント：カスタム指示は長くなりすぎると逆効果になることがあります。最も重要な指示を3～5点に絞ると、Claudeがよりきちんと守ってくれます。細かい指示は個別の会話の中で都度伝えましょう。

16.4 ナレッジを活用する

ナレッジ（Knowledge）とは、そのプロジェクトでClaudeが参考にできる文書や資料のことです。

ナレッジベース（Knowledge Base）とも呼ばれるこの機能を使うと、ファイルをアップロードしておくだけで、そのプロジェクト内の会話でいつでも内容を参照してもらえます。

16.4.1 ナレッジへのファイル追加方法

1. プロジェクトの設定画面（Project settings）を開きます
2. 「Knowledge」セクションの「Add content」または「Upload files」をクリックします
3. ファイルを選択してアップロードします
4. アップロードが完了したら、プロジェクト内の会話でそのファイルを参照できます

16.4.2 対応ファイル形式

さまざまな形式のファイルをナレッジとして登録できます。

種類	対応形式
文書	PDF、DOCX、ODT、RTF、EPUB、TXT
ウェブ・データ	HTML、CSV、JSON
表計算	XLSX
画像	JPEG、PNG、GIF、WEBP
音声	MP3、WAV

PDF に関する補足:

- 100 ページ以下の PDF は、画像・図表を含む視覚的な内容も分析できます
- 1,000 ページを超える PDF はテキスト部分のみの分析になります

16.4.3 容量制限

制限	内容
1 ファイルの最大サイズ	30 MB
1 会話への最大添付数	最大 20 ファイル



注釈:

Free プランでは、ナレッジの利用範囲はコンテキストウィンドウ（Claude が一度に処理できる情報量）の範囲内に限られます。Pro・Max・Team・Enterprise プランでは RAG（検索拡張生成 — 大量の文書から必要な情報を自動的に検索して参照する技術）により、最大 10 倍の容量のナレッジを活用できます。

16.4.4 ナレッジの活用パターン

社内マニュアル・規程を登録する

就業規則、経費精算規程、業務手順書などをアップロードしておくと、「経費精算の期限はいつですか？」「出張申請の手順を教えて」といった質問に、文書の内容をもとに答えてもらえます。

製品・サービスの資料を登録する

製品仕様書、価格表、Q&A 集などを登録しておく、顧客からの問い合わせ対応や提案資料の作成に活用できます。

研究資料・論文を登録する

調査したい文献や資料をまとめてアップロードしておく、「この論文の要点を教えて」「この3つの資料に共通している考え方を整理して」といった質問に答えてもらえます。

つまづきポイント: ナレッジに追加したファイルの内容を Claude がうまく参照してくれない場合は、「ナレッジに登録した〇〇（ファイル名）を参考に」と明示的に伝えると確実です。

16.5 活用例

業種・用途別の具体的な使い方を紹介します。プロンプト例はそのままコピーして使えます。

16.5.1 採用業務

ナレッジに「求人票」「面接評価シート」「採用ペルソナ資料」を登録しておきます。

来週の〇〇職の候補者との面接に向けて、
ナレッジに登録した求人票と評価シートをもとに
面接で使う質問リストを 10 問作ってください。

- ・候補者のバックグラウンドを確認する質問
- ・実務能力を測る質問
- ・カルチャーフィットを確認する質問

の3カテゴリに分けてください。

16.5.2 カスタマーサポート

ナレッジに「製品マニュアル」「よくある質問集 (FAQ)」「対応ガイドライン」を登録しておきます。

お客様から以下の問い合わせが来しました。
ナレッジに登録した FAQ とマニュアルを参考に、
丁寧で的確な返信メールを作成してください。

【お客様の問い合わせ内容】
(ここに問い合わせ文を貼り付ける)

返信は 300 字以内で、余計な情報は省いてください。

16.5.3 社内教育・研修資料の作成

ナレッジに「社内規程」「業務マニュアル」「過去の研修資料」を登録しておきます。

新入社員向けのオンボーディング研修の
スライド資料のアウトラインを作ってください。

- ・ 所要時間：2 時間
- ・ 対象：営業職の新入社員
- ・ 含めるべき内容：会社のルール・営業の基本フロー・CRM ツールの使い方

ナレッジに登録した業務マニュアルの内容に沿って構成してください。

16.5.4 コンテンツ制作（ブログ・SNS）

ナレッジに「過去の記事」「ブランドガイドライン」「キーワードリスト」を登録しておきます。

今月のブログ記事のテーマを 5 つ提案してください。

- ・ ターゲット読者：中小企業の経営者
- ・ 季節感：春（4 月）
- ・ ナレッジに登録した過去記事とかぶらないようにする
- ・ 各テーマには想定される検索キーワードも 1 つ添えてください

つまづきポイント：「ナレッジを活用してほしいのに、Claude が一般的な回答をしてしまう」
場合は、「ナレッジに登録した資料を参考にして」と明示的に指示に含めてみてください。

16.6 チームで共有する（Team / Enterprise）

Team プランまたは Enterprise プランでは、プロジェクトをチームメンバーと共有できます。



注釈:

チームでのプロジェクト共有は **Team プランおよび Enterprise プランのみの機能**です。Free・Pro・Max プランでは個人利用のみになります。

16.6.1 公開範囲の設定

プロジェクトには以下の公開範囲を設定できます。

設定	説明
プライベート	作成者のみアクセス可能
公開（チーム内）	組織内のすべてのメンバーがアクセス可能

16.6.2 権限の種類

メンバーごとに以下の権限を設定できます。

権限	できること
Can use (閲覧・チャット)	プロジェクト内で会話できる。カスタム指示・ナレッジの編集は不可
Can edit (編集)	カスタム指示・ナレッジの追加・変更が可能

16.6.3 メンバーの招待方法

個別に招待する場合:

1. プロジェクトの設定 (Project settings) を開きます
2. 「Share」または「Members」セクションを選択します
3. 招待したいメンバーのメールアドレスを入力します
4. 権限 (Can use / Can edit) を選択して招待します

一括で招待する場合:

複数のメンバーを招待する場合は、メールアドレスをカンマ区切りで入力するか、一覧として貼り付けることで一括招待できます。

16.6.4 チームでの活用例

カスタマーサポートチーム

- ナレッジに製品マニュアルと FAQ を登録
- カスタム指示に対応のトーンとルールを記載
- チーム全員に「Can use」権限を付与
- → 誰でも同じ品質でお客様対応ができる

マーケティングチーム

- ナレッジにブランドガイドラインとターゲット顧客の定義を登録
- カスタム指示にブランドのトーン&マナーを記載
- 担当者に「Can edit」権限、関係者に「Can use」権限
- → 一貫したブランドメッセージでコンテンツを制作できる

つまづきポイント: 共有プロジェクトのカスタム指示やナレッジを変更すると、そのプロジェクトを使っているすべてのメンバーに影響します。「Can edit」権限を付与するメンバーは慎重に選びましょう。

16.7 プランごとの違い

Projects の機能はプランによって異なります。

機能	Free	Pro / Max	Team	Enterprise
プロジェクト数	最大 5 件	無制限	無制限	無制限
カスタム指示	利用可	利用可	利用可	利用可
ナレッジベース	コンテキストウィンドウ内	RAG で最大 10 倍 拡張	RAG で最大 10 倍 拡張	RAG で最大 10 倍 拡張
チーム共有	不可	不可	利用可	利用可
管理者機能	なし	なし	あり	あり

RAG（検索拡張生成） とは、大量のナレッジ文書から必要な情報を自動的に検索して、回答に活かす技術です。Free プランではコンテキストウィンドウ（一度に処理できる情報量の上限）を超えるサイズのナレッジは活用できませんが、有料プランでは自動的に検索して参照するため、より大量の資料を登録しておけます。



警告:

Free プランの「最大 5 件」という制限は、アーカイブしたプロジェクトもカウントされます。件数の上限に達した場合は、不要なプロジェクトを削除するか、有料プランへのアップグレードを検討してください。

16.8 次のステップ

Projects の基本を理解したら、他の機能と組み合わせてさらに活用の幅を広げましょう。

- [Claude Web \(claude.ai\) の使い方](#) (ページ 89) — Projects 以外の機能 (Artifacts、Deep Research、Memory など) をまとめて確認する
- [Artifacts の使い方](#) (ページ 97) — 会話の中でドキュメントやコードを生成・編集する。Projects のナレッジと組み合わせると強力
- [Memory — Claude があなたを覚えてくれる](#) (ページ 113) — プロジェクトをまたいで Claude に覚えておいてほしい情報を管理する方法
- [Deep Research の使い方](#) (ページ 105) — 複雑な調査を Claude に任せる。調査結果をナレッジとして保存する活用法も

第17章 Claude Codeをはじめよう



注釈:

ターミナルに不慣れな方は [非エンジニア向け Claude Code ガイド](#) (ページ 137) から始めることをおすすめします。

Claude Code は、ターミナル（コマンドラインツール）上で動作する AI コーディングエージェントです。自然言語で指示するだけで、ファイルの読み書き、コードの修正、コマンドの実行、Git 操作まで、開発作業を幅広くこなしてくれます。

この記事では、Claude Code のインストールから最初のコマンド実行までを順を追って解説します。

17.1 Claude Code とは

Claude Code は、ブラウザやデスクトップアプリで使う Claude（チャット UI）とは異なる、**ターミナルネイティブの AI エージェント**です。

17.1.1 チャット版との違い

	Claude Web / Desktop	Claude Code
操作環境	ブラウザ・デスクトップアプリ	ターミナル（コマンドライン）
ファイルアクセス	アップロードが必要	ローカルファイルに直接アクセス
コマンド実行	できない	シェルコマンドを実行できる
Git 操作	できない	コミット・プッシュなど直接操作
向いている作業	質問・文章作成・チャット	コーディング・ファイル操作・自動化

Claude Code がとりわけ威力を発揮するのは、次のような場面です。

- **コードベース全体の把握:** 「このプロジェクトの構成を説明して」と頼むと、ディレクトリを走査して概要を答えてくれる
- **バグ修正:** エラーメッセージを貼り付けるだけで、原因を特定して修正案を実装してくれる
- **リファクタリング:** 「このファイルの関数を整理して」と指示すると、コードを書き換えてくれる
- **Git 操作:** 「変更内容を確認してコミットして」と頼むと、`git diff` の確認からコミットメッセージの生成まで行ってくれる
- **ファイル一括処理:** 「このフォルダ内の CSV を全部結合して」といった繰り返し作業も自動化できる

17.1.2 料金について

Claude Code は **Pro プラン以上で利用できます**。Free プランでは使えません。Claude Code の利用はチャットと同じ利用枠を共有するため、コーディング作業をメインにする場合は **Max プラン** が向いています。詳しくは **料金プランの比較** (ページ 15) を参照してください (最新の料金は [Anthropic 公式の料金ページ](#) で確認してください)。

17.2 前提条件

Claude Code をインストールする前に、以下を確認してください。

17.2.1 対応 OS

Mac

macOS 12 Monterey 以降が必要です。以下のコマンドでバージョンを確認できます。

```
sw_vers -productVersion
```

Windows

Windows でも利用できます。PowerShell、WSL、WinGet のいずれかでインストールできます (Git for Windows が必要です)。

WSL を使う場合で、まだインストールしていない場合は、PowerShell (管理者権限) で以下を実行してください。

```
wsl --install
```

17.2.2 Node.js 18 以上 (npm でインストールする場合のみ)

npm 経由で Claude Code をインストールする場合は **Node.js 18 以上**が必要です。ネイティブインストーラーを使う場合は不要です。

```
node --version
```

v18.0.0 以上が表示されれば OK です。インストールされていない、またはバージョンが古い場合は [Node.js 公式サイト](#) からインストールしてください。

つまづきポイント: node: command not found と表示される場合、Node.js がインストールされていません。[nodejs.org](#) から LTS (長期サポート版) をダウンロードしてインストールするか、Node.js が不要なネイティブインストーラーの利用を検討してください。

17.2.3 Git

Git 操作の機能を使うには Git が必要です。

```
git --version
```

インストールされていない場合は [Git 公式サイト](#) からインストールできます。

Mac

macOS では Xcode Command Line Tools をインストールすると Git も含まれます。

```
xcode-select --install
```

Windows

WSL 環境では以下でインストールできます。

```
sudo apt-get update && sudo apt-get install git
```

17.3 インストール

Claude Code のインストール方法は複数あります。ネイティブインストーラーが公式推奨の方法です。

17.3.1 ネイティブインストーラー（推奨）

Node.js 不要で、バックグラウンド自動更新にも対応しています。

Mac

```
curl -fsSL https://claude.ai/install.sh | bash
```

Windows

PowerShell の場合:

```
irm https://claude.ai/install.ps1 | iex
```

WinGet の場合:

```
winget install Anthropic.ClaudeCode
```

インストールが完了したら、以下でバージョンを確認します。

```
claude --version
```

バージョン番号が表示されれば成功です。

17.3.2 Homebrew でインストール（macOS のみ）

Mac

Homebrew を使っている場合は以下でもインストールできます。

```
brew install --cask claude-code
```

Windows

Windows ではネイティブインストーラーまたは npm を使ってインストールしてください。

17.3.3 npm でインストール

Node.js 18 以上が必要です。ネイティブインストーラーの利用が難しい場合に使えます。

```
npm install -g @anthropic-ai/claude-code
```

つまづきポイント: npm: `command not found` と表示される場合、Node.js がインストールされていません。前のセクションの「Node.js 18 以上」を先に対応してください。

EACCES: `permission denied` エラーが出た場合は、`sudo npm install -g @anthropic-ai/claude-code` を試してください。ただし、`sudo` での npm グローバルインストールは権限やセキュリティの問題が生じる場合があるため、公式はネイティブインストーラーの利用を推奨しています。

17.4 認証 (ログイン)

インストール後、はじめて `claude` コマンドを実行すると認証フローが始まります。

```
claude
```

初回起動時には、Anthropic アカウントへの認証を求めるメッセージが表示されます。ブラウザが自動的に開き、`claude.ai` の認証ページに移動します。

17.4.1 認証の手順

1. ターミナルで `claude` を実行する
2. 自動的にブラウザが開き、Anthropic の認証ページが表示される
3. `claude.ai` にログインする (アカウントがない場合はサインアップする)
4. 認証が完了すると、ブラウザに「認証成功」のメッセージが表示される
5. ターミナルに戻ると、Claude Code が起動している

つまづきポイント: ブラウザが自動的に開かない場合は、ターミナルに表示される URL をコピーしてブラウザに貼り付けてください。

重要: Claude Code を使うには **Pro プラン以上**のサブスクリプションが必要です。Free プランのアカウントでログインすると、利用できない旨のエラーが表示されます。

17.4.2 API キーで認証する場合

Anthropic の API キー (有料の API アクセス) を持っている場合は、環境変数でも認証できます。

```
export ANTHROPIC_API_KEY="your-api-key-here"
claude
```

17.5 CLAUDE.md の作成

CLAUDE.md は、プロジェクトのルートディレクトリに置く設定ファイルです。このファイルに書いた内容は、Claude Code がそのプロジェクト内で作業する際に自動的に読み込まれ、プロジェクトの文脈として考慮されます。

毎回「このプロジェクトは Next.js で書かれていて...」と説明する手間がなくなります。

17.5.1 CLAUDE.md のテンプレート

プロジェクトのルートで以下のファイルを作成します。

```
touch CLAUDE.md
```

以下はテンプレートです。プロジェクトに合わせて書き換えてください。

プロジェクト概要

このプロジェクトは [プロジェクトの説明を 1~2 文で] です。

技術スタック

- 言語: TypeScript
- フレームワーク: Next.js 15 (App Router)
- データベース: PostgreSQL
- パッケージマネージャー: pnpm

ディレクトリ構成

- `src/app/` — ページコンポーネント
- `src/components/` — 共通コンポーネント
- `src/lib/` — ユーティリティ関数
- `prisma/` — データベーススキーマ

コーディング規約

- TypeScript の strict モードを使用する
- コンポーネントは関数型コンポーネントで書く
- テストは Vitest を使用する
- コミットメッセージは Conventional Commits に従う

よく使うコマンド

- `pnpm dev` — 開発サーバー起動
- `pnpm build` — 本番ビルド
- `pnpm test` — テスト実行
- `pnpm lint` — Lint チェック

(次のページに続く)

(前のページからの続き)

注意点

- ``main`` ブランチへの直接プッシュは禁止。必ず Pull Request を作成すること
- 環境変数は ``.env.local`` に書き、``.env.example`` に変数名だけ記録すること

CLAUDE.md はプロジェクトの規模や種類に合わせて自由にカスタマイズできます。書けば書くほど Claude Code の回答精度が上がりますが、最初はシンプルなものから始めて徐々に育てていくのがおすすめです。

Tips: CLAUDE.md はホームディレクトリ (`~/CLAUDE.md`) にも置けます。全プロジェクト共通の設定 (好みのコードスタイルや言語設定など) を書いておくと便利です。

17.6 基本的な使い方

Claude Code には 3 つの主な使い方があります。

17.6.1 1. 対話モード (会話しながら作業)

ターミナルで `claude` と入力するだけで、対話型のシェルが起動します。

```
claude
```

起動すると `>` プロンプトが表示されます。ここに自然言語で指示を入力します。

```
> このプロジェクトの構成を説明して
> src/app/page.tsx のコードをレビューして
> README.md を更新して
> git の差分を確認してコミットして
```

複数のタスクを会話形式で進めたいときや、結果を確認しながら作業を深めたいときに向いています。終了するには `Ctrl + C` または `/exit` と入力します。

17.6.2 2. ワンショットモード (1 回だけ実行)

`-p` オプションを使うと、1 つの指示を実行してすぐに終了します。

```
claude -p "このプロジェクトの README を生成して"
claude -p "src ディレクトリ内のすべての TypeScript ファイルの一覧を表示して"
claude -p "最新のコミットの内容を日本語で説明して"
```

スクリプトへの組み込みや、定型タスクの自動化に便利です。

17.6.3 3. パイプで入力を渡す

他のコマンドの出力を Claude Code に渡すこともできます。

```
# ファイルの内容をレビューしてもらう
cat src/utils/format.ts | claude -p "このコードをレビューして改善点を教えて"

# エラーログを解析してもらう
cat error.log | claude -p "このエラーログを解析して原因を特定して"

# git diff の内容を確認してもらう
git diff | claude -p "この変更の概要をコミットメッセージ形式で書いて"
```

つまづきポイント: 対話モードで長い会話を続けると、コンテキスト（AI が覚えられる情報の量）の上限に近づくことがあります。その場合は `/compact` コマンドで会話を要約するか、`/clear` で会話をリセットしましょう。

17.7 主要なスラッシュコマンド

対話モード中に使えるスラッシュコマンドの一覧です。

コマンド	説明
<code>/help</code>	使えるコマンドの一覧を表示する
<code>/clear</code>	会話の履歴をリセットする（新しいタスクを始めるとき）
<code>/compact</code>	会話を要約してコンテキストを節約する
<code>/cost</code>	現在のセッションで使ったトークン数とコストを表示する
<code>/model</code>	使用するモデルを切り替える（Sonnet / Opus など）
<code>/exit</code>	Claude Code を終了する
<code>/bug</code>	バグや問題を Anthropic に報告する
<code>/review</code>	最後のレスポンスを評価する（フィードバック送信）

17.7.1 コンテキストのコントロール

Claude Code が参照するファイルを明示的に指定することもできます。

```
# 特定のファイルを読み込んで起動
claude --file src/components/Button.tsx

# ディレクトリを指定して起動
claude --dir src/components
```

対話中に `@ファイルパス` と入力すると、そのファイルをコンテキストに追加できます。

```
> @src/api/users.ts このファイルの型定義を整理して
```

Tips: 大規模なプロジェクトでは、最初に `>` このプロジェクトの `CLAUDE.md` を読んで、構成

を把握して と頼むと、その後の会話の精度が上がります。

17.8 実際に試してみる

インストールが完了したら、実際に動かしてみましょう。既存のプロジェクトがある場合はそのディレクトリで、ない場合はどこか適当なフォルダで試せます。

```
# プロジェクトのディレクトリに移動
cd your-project

# Claude Code を起動
claude

# 試してみるコマンド例
> 今いるディレクトリの構成を説明して
> README.md があれば読んで内容を要約して
> git log の最新 10 件を日本語で説明して
```

最初の一步として、Claude Code に**現在のプロジェクトを把握してもらう**ことから始めるのがおすすめです。

17.9 次のステップ

Claude Code の基本的な使い方を理解できたら、さらにステップアップしましょう。

- **非エンジニア向け Claude Code ガイド** (ページ 137) — ターミナル未経験でも大丈夫。やさしい入門ガイド
- **Claude Code 実践ワークフロー集** (ページ 147) — コードレビュー、バグ修正、リファクタリングなど実践的なワークフロー
- **IDE 統合** (ページ 159) — VS Code や JetBrains IDEs と Claude Code を組み合わせて使う方法
- **料金プランの比較** (ページ 15) — Pro / Max プランの違いと Claude Code の利用量

第18章 非エンジニア向け Claude Code ガイド

「Claude Code ってエンジニア向けでしょ？」——そう思っていないですか？

実は Claude Code は、プログラミングを知らなくても使える場面がたくさんあります。大量のファイルを一括整理したい、CSV の集計を自動化したい、フォルダ内の画像ファイル名をまとめて変更したい——そんな日常業務の自動化に、Claude Code はとても役立ちます。

このガイドでは、ターミナル（黒い画面）を触ったことのない方を対象に、Claude Code のインストールから実際のタスク実行まで、一つひとつ丁寧に解説します。

18.1 なぜ非エンジニアにも Claude Code が役立つのか

Claude には複数の使い方があります。ブラウザで使う **Claude Web**、デスクトップアプリの **Claude Desktop**、そして今回紹介する **Claude Code** です。それぞれ得意なことが異なります。

	Claude Web / Desktop	Claude Code
操作方法	ブラウザ・アプリ	ターミナル（黒い画面）
ファイル操作	1 ファイルずつ手で添付	フォルダごと自動で操作
一括処理	苦手	得意（何百ファイルでも）
自動化・繰り返し作業	毎回手動	一度設定すれば繰り返し実行
インターネット接続	必要	必要（API 通信で動作）

Claude Code が特に向いているケース：

- フォルダ内の **100 個のファイル**をまとめて 名前変更したい
- 毎週受け取る **CSV ファイル**を自動で集計 してレポート化したい
- 複数の Word ファイルから **特定の情報を一括抽出** したい
- 画像ファイルの **リサイズや形式変換を大量に** 処理したい
- ローカルに保存されたファイルを **ブラウザで開かずに処理** したい

一方で、「このメールの返信文を書いて」「この文章を要約して」といった 1 回限りの質問 には、ブラウザで使える Claude Web の方が手軽です。どちらが自分に合っているかは、記事の後半の [Cowork との使い分け](#) (ページ 145) を参照してください。

18.2 ターミナルは怖くない — 3分でわかる基礎知識

Claude Code を使うには「ターミナル」というアプリを使います。「黒い画面に文字を打ち込む」——そのイメージから怖いと感じる方も多いですが、大丈夫です。**間違ったコマンドを打ってもパソコンが壊れることはありません。**

18.2.1 ターミナルとは？

ターミナルとは、パソコンへの**文字で命令を伝えるアプリ**です。

普段はマウスでクリックしてアプリを操作しますよね。ターミナルでは同じことを「文字」で行います。たとえば「デスクトップにあるフォルダを開く」という操作をターミナルで行うと、こんな感じになります。

```
cd ~/Desktop/プロジェクトフォルダ
```

これだけです。難しそうに見えますが、Claude Code を使うなら **ほとんどの場合このような入力是不要**です。Claude に自然な日本語で「デスクトップのプロジェクトフォルダを開いて」と話しかければ、Claude が代わりにコマンドを実行してくれます。

18.2.2 コマンドとは？

コマンドとは、パソコンへの「命令文」です。英語の単語や記号を組み合わせたもので、Enter キーを押すと実行されます。

最初に覚えるコマンドは 2 つだけ:

コマンド	意味
claude	Claude Code を起動する
exit	終了する

これだけ知っていれば、Claude Code は始められます。あとは日本語で話しかけるだけです。

18.2.3 ディレクトリ（フォルダ）とは？

ターミナルでは「フォルダ」のことを「ディレクトリ」と呼ぶことがあります。意味はまったく同じです。ターミナルの画面で「ディレクトリ」と出てきたら「フォルダのことだ」と思えば OK です。

安心してください: ターミナルで何かを打ち間違えても、Enter を押さない限り何も起きません。打ち間違えたら Backspace キーで消して打ち直すだけです。もし不安になったら、Ctrl + C を押すと作業を中断できます。

18.3 インストール手順（丁寧版）

18.3.1 ターミナル不要の方法: ブラウザから使う

実は、ターミナルを一切使わずにブラウザだけで Claude Code を使う方法もあります。claude.ai/code にアクセスするだけで、インストール作業なしに Claude Code の機能を試せます。

ターミナルのインストールが難しいと感じた方や、まず手軽に試してみたい方は、こちらのブラウザ版から始めてみてください。

以下は、ターミナル版の Claude Code をインストールする手順です。ブラウザ版では物足りなくなったときや、ローカルファイルを直接操作したい場合に進めてください。

Claude Code を使うには、以下の2つをインストールする必要があります。

1. **Node.js**（Claude Code を動かすための土台）
2. **Claude Code**（本体）

難しそうに感じるかもしれませんが、手順通りに進めれば10～15分で完了します。一緒に進めましょう。

18.3.2 ステップ1: Node.js をインストールする

Node.js（ノード・ジェイエス）とは、Claude Code を動かすために必要なソフトウェアです。Claude Code 専用ではなく、さまざまなアプリの土台として広く使われている安全なソフトウェアです。

Mac

macOS の場合

1. <https://nodejs.org/ja/> をブラウザで開きます
2. 「**LTS 推奨版**」と書かれた緑のボタンをクリックしてダウンロードします（LTS は「長期サポート版」の意味で、安定して使えるバージョンです）
3. ダウンロードされた **.pkg** ファイルをダブルクリックして開きます
4. インストーラーの指示に従って「続ける」「インストール」をクリックします（パスワードの入力を求められる場合があります）
5. 「インストールが完了しました」と表示されたら完了です

インストールの確認:

インストールが完了したら、ターミナルアプリを開いて確認します。

- Spotlight 検索（Command + Space）で「ターミナル」と入力してアプリを開く
- 以下を入力して Enter を押す:

```
node --version
```

v22.0.0 のようなバージョン番号が表示されれば成功です。

Windows

Windows の場合

Windows では、Claude Code を使うために **WSL**（Windows Subsystem for Linux の略）という仕組みが必要です。「Windows の中に Linux という別の OS の環境を作る」ものですが、難しく考えなくて大丈夫です。手順通りに進めれば自動でセットアップされます。

WSL のインストール:

1. キーボードの Windows キーを押してスタートメニューを開く

2. 「**PowerShell**」と検索し、「**管理者として実行**」をクリックする

3. 以下のコマンドを入力して Enter を押す:

```
wsl --install
```

4. インストールが完了したらパソコンを再起動します

5. 再起動後、「Ubuntu」アプリが自動的に開きます。ユーザー名とパスワードを設定します（このパスワードはこの Linux 環境用のもので、Windows のパスワードとは別です）

Ubuntu に Node.js をインストール:

Ubuntu（WSL の画面）で以下を順番に入力します。各行を入力したら Enter を押してください。

```
curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -
```

```
sudo apt-get install -y nodejs
```

途中でパスワードを聞かれたら、先ほど設定したパスワードを入力します（入力しても画面には何も表示されませんが、正しく入力されています）。

確認:

```
node --version
```

v22.0.0 のようなバージョン番号が表示されれば成功です。

つまづきポイント: 「WSL --install がエラーになる」という場合、Windows のバージョンが古い可能性があります。Windows Update で最新版にアップデートしてから再試行してください。Windows 10 バージョン 2004 以降、または Windows 11 が必要です。

18.3.3 ステップ 2: Claude Code をインストールする

Node.js がインストールできたら、次は Claude Code 本体をインストールします。

Mac

ターミナルで以下のコマンドを入力して Enter を押します:

```
npm install -g @anthropic-ai/claude-code
```

インストールには数分かかります。進行状況がテキストで表示されるので、完了するまで待ちましょう。

Windows

WSL（Ubuntu）の画面で以下のコマンドを入力して Enter を押します:

```
npm install -g @anthropic-ai/claude-code
```

インストールには数分かかります。完了するまで待ちましょう。

18.3.4 ステップ 3: ログインする

Claude Code を初めて起動すると、Anthropic アカウントへのログインが求められます。

Mac

ターミナルで以下を入力して Enter を押します:

```
claude
```

Windows

WSL (Ubuntu) の画面で以下を入力して Enter を押します:

```
claude
```

初回起動時にブラウザが開き、ログイン画面が表示されます。Anthropic アカウント (claude.ai で使っているアカウント) でログインしてください。

ログインが完了するとターミナルに **Welcome to Claude Code!** と表示され、チャット入力待ちの状態になります。

料金について: Claude Code の利用には **Claude Pro プラン以上** (月額 \$20~) のサブスクリプションが必要です。無料プランでは利用できません。料金の詳細は [料金プラン比較](#) (ページ 15) を参照してください。

つまづきポイント: 「`npm: command not found` と表示される」場合、Node.js のインストールが完了していない可能性があります。ターミナルを一度閉じて再度開き、`node --version` で確認してみてください。

18.4 非エンジニアのための活用タスク 5 選

Claude Code が起動したら、日本語でそのまま話しかけるだけです。「〇〇してください」と伝えれば、Claude が必要な作業を自動で行ってくれます。

以下は、コーディング不要で使える具体的なタスク例です。

18.4.1 タスク 1: フォルダの中身を整理する

場面: デスクトップが散らかって收拾がつかなくなったとき

Claude Code に話しかける例:

```
デスクトップの「ダウンロード整理」フォルダの中のファイルを、  
拡張子ごとにサブフォルダに分けて整理してください。  
PDF は「PDF」、画像 (jpg, png) は「画像」、Excel は「Excel」フォルダに移動してください。
```

Claude は現在のフォルダ構成を確認し、移動先フォルダを作成してファイルを整理します。実行前に「このような操作を行います。よろしいですか?」と確認を求めているので、内容を見てから許可できます。

18.4.2 タスク 2: CSV ファイルを集計する

場面: 毎月の売上データを集計してレポートにまとめたいとき

Claude Code に話しかける例:

デスクトップにある「2025 年売上データ.csv」を開いて、
商品カテゴリ別の月次売上合計を計算し、
結果を「売上集計.csv」として保存してください。

Claude は CSV ファイルの内容を読み取り、集計処理を行って新しいファイルを作成します。Excel を開かなくても、大量のデータを素早く集計できます。

18.4.3 タスク 3: Word・PDF ファイルを要約する

場面: 大量の資料を読まなければならないとき

Claude Code に話しかける例:

「契約書フォルダ」の中にある PDF ファイルをすべて読んで、
各ファイルの要点を 3 行ずつまとめた一覧表を作成してください。
結果は「契約書サマリー.txt」に保存してください。

フォルダ内の複数ファイルをまとめて処理できるのは、Claude Code の大きな強みです。Web 版では 1 ファイルずつ手動で添付する必要がありますが、Claude Code はフォルダごと指定して一括処理できます。

18.4.4 タスク 4: ファイル名を一括変更する

場面: 写真や資料のファイル名をまとめて変えたいとき

Claude Code に話しかける例:

「写真 2024」フォルダの中にある画像ファイルのファイル名を、
「20241001_001.jpg」「20241001_002.jpg」のように
日付+連番の形式に一括変更してください。
元のファイル名の日付情報（ファイルの更新日）を使ってください。

100 枚・200 枚の写真でも、数秒~数十秒で処理が完了します。

18.4.5 タスク 5: 英語の PDF を日本語に翻訳する

場面: 英語の資料を日本語で読みたいとき

Claude Code に話しかける例:

デスクトップにある「英語レポート.pdf」を日本語に翻訳して、
「日本語レポート.txt」として保存してください。
専門用語はなるべく日本のビジネス現場で使われる表現にしてください。

つまづきポイント: 「Claude が処理を開始したが、途中で止まってしまった」場合は、Ctrl + C で中断して、もう少し具体的な指示に言い直してみてください。「どのフォルダにあるか」「どんな形式で保存するか」を明記すると、スムーズに動くことが多いです。

18.5 安全に使うためのポイント

Claude Code はローカルのファイルを直接操作するため、「間違っって大切なファイルを消してしまったら？」と不安になる方もいるでしょう。安心してください。Claude Code にはいくつかの安全策が組み込まれています。

18.5.1 Claude は操作前に確認を求める

Claude Code はファイルの削除・移動・上書きなど、元に戻しにくい操作を行う前に必ず確認を求めます。

以下の操作を実行します：

- 「古いファイル」フォルダ内の 52 個のファイルを削除します

実行してよいですか？ (y/n)

y と入力して Enter を押すと実行、n と入力すると中止です。不安なら n を押して指示を変えましょう。

18.5.2 重要なファイルは事前にバックアップを

Claude Code に重要なフォルダを操作させる前に、別の場所にコピーを作っておくと安心です。

「プロジェクトフォルダ」の内容を「バックアップ_20250101」という名前でデスクトップにコピーしてから、整理作業を始めてください。

こうお願いすれば、Claude がバックアップを作ってから作業を開始してくれます。

18.5.3 操作を途中で止める

作業の途中で「やっぱり止めたい」と思ったら、Ctrl + C を押してください。実行中の処理を即座に中断できます。

安心メッセージ: Ctrl + C はいつでも押せる「緊急停止ボタン」です。怖いと感じたらすぐ押してください。処理が止まるだけで、パソコンに何か悪いことが起きるわけではありません。

18.5.4 Permission (許可) の仕組み

Claude Code は初回起動時に「このフォルダへのアクセスを許可しますか？」と確認します。アクセスを許可したフォルダ以外のファイルには触れません。最初は作業したいフォルダだけを許可するのが安全です。

18.6 よくある不安 Q&A

18.6.1 Q. ファイルが壊れませんか？

Claude Code はファイルを「読む」「コピーする」「移動する」「新しく作る」という操作が中心です。ファイルの削除や上書きを行う場合は必ず事前に確認を求めます。万が一のために、重要なファイルは事前にバックアップを取っておくことをお勧めします。

18.6.2 Q. 間違った操作をしたら取り消せますか？

Claude Code は削除確認を必ず行いますが、万全を期するなら**バックアップを取ってから作業する**のが最も安全です。また、Claude に「さっきの操作を元に戻してください」と伝えると、可能な範囲で元に戻してくれます。

18.6.3 Q. 料金はいくらかかりますか？

Claude Code の利用には **Claude Pro プラン**（月額 \$20）以上が必要です。無料プランでは Claude Code を利用できません。詳しくは **料金プラン比較**（ページ 15）を参照してください。

18.6.4 Q. Web 版（claude.ai）と何が違うのですか？

	Claude Web	Claude Code
操作画面	ブラウザ	ターミナル
ファイル操作	1 ファイルずつ手動添付	フォルダごと自動で操作
一括処理	難しい	得意
会話のしやすさ	しやすい	やや慣れが必要
インストール	不要	必要

一言で言うと：日常的な質問・文書作成は Claude Web、大量ファイルの自動処理は Claude Code、という使い分けがおすすめです。

18.6.5 Q. 操作中にターミナルが固まったように見えます

処理に時間がかかっているときは、ターミナルが無反応に見えることがあります。1～2 分待ってみてください。それでも変化がなければ Ctrl + C で中断して、もう一度試してみましょう。

18.6.6 Q. 「Permission denied」と表示されました

アクセスが許可されていないフォルダを操作しようとしたときに表示されます。Claude に「まず〇〇フォルダへのアクセスを許可してください」と伝えるか、対象フォルダの場所を確認してから再試行してください。

18.7 Cowork との使い分け

Claude Desktop の **Cowork** も、ファイル操作やタスク自動化ができる機能です。「Claude Code と何が違うの？」という疑問にお答えします。

観点	Claude Code	Cowork
操作方法	ターミナル（文字入力）	デスクトップアプリ（GUI）
インストール	Node.js + npm	Claude Desktop アプリ
対象ユーザー	ターミナルに慣れた人	ターミナルを使いたくない人
外部サービス連携	設定が必要	Slack・Notion 等と連携しやすい
繰り返し自動化	スクリプト化が必要	スケジュール設定が簡単
ファイル操作の柔軟性	高い	高い

どちらを選ぶか迷ったら、以下を参考にしてください:

- ターミナルに抵抗がある → Cowork
- Slack・Notion・Gmail と自動連携したい → Cowork
- 大量のファイルを細かい条件で処理したい → Claude Code
- プログラムに近い操作（条件分岐、繰り返し処理）が必要 → Claude Code
- どちらか一方だけ使う必要はない → 場面に応じて両方使うのもアリです

つまづきポイント:「どちらを使うか」で悩んで始められない方が多いです。まずは Claude Code を試してみて、「やっぱりターミナルは難しい」と感じたら Cowork に切り替えるという順番でも問題ありません。両方試してみて、自分に合った方を選びましょう。

18.8 次のステップ

Claude Code の基本を理解できましたか？さらに活用の幅を広げるために、以下の記事もあわせて読んでみてください。

インストールの詳細・基本コマンドをもっと知りたい方:

- [Claude Code をはじめよう](#) (ページ 129) — インストール手順のより詳しい解説と基本コマンド一覧

実際の業務での使い方を学びたい方:

- [Claude Code 実践ワークフロー集](#) (ページ 147) — ファイル整理・データ処理・ドキュメント生成などの業務別レシピ

ターミナルを使わずにタスク自動化したい方:

- [Cowork 入門 — Claude にタスクを任せよう](#) (ページ 179) — GUI で使えるタスク自動化ツールの使い方

Claude の料金・プランを確認したい方:

- [Claude の料金プラン比較](#) (ページ 15) — Free・Pro・Max の違いと Claude Code が使えるプランを確認する

第19章 Claude Code 実践ワークフロー集

Claude Code を「何かわからないことがあったときに質問する」ツールとして使っていませんか？実は Claude Code は、**作業フロー全体に組み込む**ことで本来の力を発揮します。

この記事では、開発者から非エンジニアまで幅広く使える実践的なワークフローを紹介します。それぞれ「いつ使うか」「具体的な手順」「コピペ可能なコマンド例」を統一フォーマットでまとめています。

19.1 ワークフローの考え方

Claude Code を使いこなすポイントは2つあります。

19.1.1 単発の質問より「フロー全体への組み込み」

多くの人は「このバグを直して」「このコードを説明して」という**単発の質問**から始めます。これでも十分便利ですが、Claude Code の真価は**作業の流れ全体に組み込んだとき**に発揮されます。

たとえばコードレビューのワークフローを考えてみましょう。

単発の使い方（効率が低い）：

```
# 毎回手動でファイルを指定して質問する
> src/components/Button.tsx を確認してください
> では src/hooks/useAuth.ts も確認してください
```

フロー全体への組み込み（効率が高い）：

```
# PR の差分全体を一度にレビューさせる
git diff main | claude -p "この変更をレビューして、問題点をリスト形式でまとめて"
```

19.1.2 CLAUDE.md で文脈を設定する

Claude Code は**プロジェクトの文脈を知らない状態で起動します**。毎回「このプロジェクトは TypeScript で書かれていて、テストは Vitest を使っていて...」と説明するのは非効率です。

CLAUDE.md をプロジェクトルートに置くと、Claude Code が起動するたびに自動的に読み込まれます。一度書いておけば、今後の会話で毎回説明する必要がなくなります。

```
# プロジェクトルートに CLAUDE.md を作成
touch CLAUDE.md
```

CLAUDE.md に書くべき内容：

プロジェクト概要

このプロジェクトは〇〇のための Web アプリです。

技術スタック

- 言語: TypeScript
- フレームワーク: Next.js 15 (App Router)
- テスト: Vitest + Testing Library
- パッケージマネージャー: pnpm

コーディング規約

- コンポーネントは関数型で書く
- 型は明示的に定義する (any を避ける)
- テストファイルは `__tests__` ディレクトリに置く

よく使うコマンド

- ``pnpm dev`` — 開発サーバー
- ``pnpm test`` — テスト実行
- ``pnpm build`` — 本番ビルド

Tips: CLAUDE.md は育てるものです。作業しながら「毎回同じ説明をしているな」と気づいたら、その内容を CLAUDE.md に追記していきましょう。半月ほどでかなり賢い指示書になります。

19.1.3 対話モードとワンショットモードの使い分け

モード	コマンド	向いている場面
対話モード	<code>claude</code>	結果を確認しながら進めたい、複数ステップの作業
ワンショットモード	<code>claude -p "..."</code>	1回で完結するタスク、スクリプトへの組み込み
パイプモード	コマンド <code>claude -p "..."</code>	別コマンドの出力を解析・加工したい

19.2 コードレビュー

難易度: ★☆☆ (初級)

19.2.1 いつ使うか

- PR を出す前に自分でセルフレビューしたいとき
- レビューアが捕まらないが早めに品質確認したいとき
- コードの問題点を網羅的に洗い出したいとき

19.2.2 手順

パターン 1: git diff を使ったワンショットレビュー（推奨）

```
# main ブランチとの差分を全部レビューさせる
git diff main | claude -p "このコードの変更をレビューして。バグの可能性、パフォーマンス上の懸念、可読性の問題を指摘してリスト形式でまとめて"
```

```
# ステージング済みの変更だけをレビューする
git diff --staged | claude -p "このコードの変更をレビューして、問題点を箇条書きで教えて"
```

パターン 2: 特定ファイルをレビューする

```
# 対話モードで特定ファイルを詳しくレビュー
claude
> @src/lib/auth.ts このファイルをレビューして。特にセキュリティ上の問題がないか確認して
```

パターン 3: PR のコミット履歴を元にレビューコメントを生成する

```
# 最新コミットのメッセージを含めてレビューさせる
git log --oneline -10 | claude -p "これらのコミットに含まれる変更について、レビュアーが確認すべきポイントを教えて"
```

19.2.3 Tips

つまづきポイント: `git diff main` を実行したときに差分が大量になってしまう場合は、ファイルを絞り込む方法が有効です。 `git diff main -- src/components/` のようにディレクトリを絞るか、対話モードで重要なファイルを順番にレビューしてもらいましょう。

コミットメッセージを自動生成するワンショット：

```
# git diff の内容からコミットメッセージを生成する
git diff --staged | claude -p "この変更の内容を Conventional Commits 形式のコミットメッセージにまとめて（1行で）"
```

19.3 バグ修正

難易度: ★☆☆（初級）

19.3.1 いつ使うか

- エラーメッセージの意味がわからないとき
- スタックトレースをどう読めばいいかわからないとき
- 原因は特定できたが修正方法がわからないとき

19.3.2 手順

パターン 1: エラーログをそのまま貼り付ける（最も手軽）

```
# エラーログをパイプで渡す
cat error.log | claude -p "このエラーの原因を特定して、修正方法を教えて"
```

```
# コマンドの実行結果とエラーと一緒に渡す
pnpm build 2>&1 | claude -p "このビルドエラーを解析して、どのファイルの何行目を修正すればいいか教えて"
```

パターン 2: 対話モードでデバッグを進める

```
claude
> TypeError: Cannot read properties of undefined (reading 'map') というエラーが出ています
> スタックトレースはこちらです: [スタックトレースを貼り付ける]
> @src/components/UserList.tsx このファイルを確認して原因を特定して
```

パターン 3: テストを使ってバグを再現・修正する

```
claude
> pnpm test を実行したら以下のテストが失敗しました
> [テスト失敗のログを貼り付ける]
> テストが通るように修正してください
```

19.3.3 Tips

つまづきポイント:「エラーが出ています」だけでは Claude に情報が少なすぎます。**エラーメッセージ全文**と**どの操作をしたときに発生するか**をセットで伝えると、格段に精度が上がります。「ログインボタンをクリックしたときに以下のエラーが出ます」という形式が理想的です。

再現手順を明示した指示の例：

```
> 以下の状況でエラーが発生します：
> 1. http://localhost:3000/dashboard にアクセス
> 2. ユーザー名とパスワードを入力してログインボタンをクリック
> 3. 以下のエラーが発生する：
>   [エラーメッセージを貼り付け]
>
> @src/app/api/auth/route.ts と @src/lib/session.ts を確認して、原因を特定して修正してください
```

19.4 リファクタリング

難易度: ★★☆☆ (中級)

19.4.1 いつ使うか

- コードが複雑になってきて整理したいとき
- 関数やコンポーネントが大きくなりすぎたとき
- 技術的負債を段階的に解消したいとき
- 大規模リファクタリングを安全に進めたいとき

19.4.2 手順

パターン 1: 単一ファイルの整理 (小規模)

```
claude
> @src/utils/data-transform.ts このファイルをリファクタリングしてください
> 目標:
> - 1つの関数が1つのことだけをやるように分割する
> - 変数名をわかりやすくする
> - 重複するロジックをまとめる
> テストが通ることを確認しながら進めてください
```

パターン 2: 大規模リファクタリングの分割戦略

大規模なリファクタリングを一度にやろうとすると、予期せぬ問題が多発します。**段階的に進める**ことが重要です。

まずリファクタリング計画を立てさせます:

```
claude
> src/services/ ディレクトリ全体のコードを整理したいと思っています
> まずは現状の問題点を分析して、どの順番でリファクタリングを進めるべきか計画を立ててください
> 各ステップは独立してマージできるサイズに分割してください
```

計画が決まったら、1ステップずつ進めます:

```
claude
> 先ほどの計画のステップ1 (UserService の分割) を実装してください
> 変更後はテストを実行して、既存のテストが全て通ることを確認してください
```

パターン 3: パフォーマンス改善

```
# 特定の処理が遅い場合
cat src/lib/heavy-computation.ts | claude -p "このコードのパフォーマンス上のボトルネックを特定して、改善案を示してください"
```

19.4.3 Tips

つまづきポイント: リファクタリング中に「これも直したい」「あれも気になる」と範囲を広げてしまうと、收拾がつかなくなります。「今回のスコープはこのファイルだけ」と明示的に伝えることが重要です。Claude も同様の問題を起こすことがあるので、「このファイルだけを変更してください。他のファイルには触れないでください」と指示しましょう。

リファクタリング前後の確認フロー：

```
# 1. リファクタリング前にテストを確認
pnpm test

# 2. リファクタリングを実行
claude
> @src/components/Form.tsx をリファクタリングしてください

# 3. リファクタリング後にテストを再実行して確認
pnpm test
```

19.5 テスト生成

難易度: ★★☆☆ (中級)

19.5.1 いつ使うか

- テストを書くのが後回しになっているコードがあるとき
- 新機能を実装したあとにテストを追加したいとき
- テストカバレッジを上げたいとき
- どんなテストを書けばいいかわからないとき

19.5.2 手順

パターン 1: 既存の関数に対するユニットテストを生成する

```
claude
> @src/lib/format.ts このファイルの全関数に対するユニットテストを生成してください
> - テストフレームワーク: Vitest
> - テストファイルの保存先: src/lib/__tests__/format.test.ts
> - エッジケース (空文字列、null、undefined、境界値) も含めてください
```

パターン 2: React コンポーネントのテストを生成する

```
claude
> @src/components/SearchInput.tsx このコンポーネントのテストを生成してください
> - Testing Library を使ってください
```

(次のページに続く)

(前のページからの続き)

```
> - ユーザー操作（入力、クリア）のテストを含めてください
> - アクセシビリティの確認（aria 属性など）もテストしてください
```

パターン 3: API エンドポイントのテストを生成する

```
claude
> @src/app/api/users/route.ts この API エンドポイントのテストを生成してください
> - 正常系（200 レスポンス）と異常系（400、401、500）のケースを含めてください
> - モックが必要なデータベース呼び出しは適切にモックしてください
```

パターン 4: テストカバレッジを確認して足りない部分を補う

```
# カバレッジレポートを生成
pnpm test --coverage 2>&1 | claude -p "このカバレッジレポートを見て、テストが不足している重要な部分を特定して、追加すべきテストケースを教えてください"
```

19.5.3 Tips

つまづきポイント: 生成されたテストをそのまま使うと、「テストのためのテスト」（実際の動作を確認していないテスト）になることがあります。生成後は必ずテストの内容を確認し、「このテストは何を確認しているか」を理解してから使いましょう。Claude に「このテストが何を確認しているか説明して」と聞くと理解が深まります。

テスト生成後の確認手順：

```
# 生成したテストを実行して通ることを確認
pnpm test src/lib/__tests__/format.test.ts

# テストの説明を求める
claude
> 生成したテストのうち、一番複雑なテストケースを説明してください
```

19.6 ドキュメント生成

難易度: ★☆☆（初級）

19.6.1 いつ使うか

- コードの説明が不足していてチームメンバーが理解しにくいとき
- API を公開するために仕様書を作りたいとき
- README が古くなっているとき
- JSDoc コメントを一括で追加したいとき

19.6.2 手順

パターン 1: README を自動生成する

```
claude -p "このプロジェクトの README.md を生成してください。以下を含めてください：概要、インストール方法、使い方、設定オプション、ライセンス"
```

または既存の README を更新する場合：

```
claude
> @README.md を読んで、現在の状態を確認してください
> 次に、実際のプロジェクト構成（ディレクトリ構造、package.json）を確認して、README.md の内容が古くなっている箇所を特定してください
> 最新の状態に更新してください
```

パターン 2: 関数・クラスに JSDoc コメントを追加する

```
claude
> @src/lib/api-client.ts このファイルの全ての関数・クラスに JSDoc コメントを追加してください
> - パラメータの型と説明
> - 戻り値の型と説明
> - 使用例 (@example)
> を含めてください。日本語で書いてください
```

パターン 3: API ドキュメントを生成する

```
claude
> src/app/api/ ディレクトリを調べて、全 API エンドポイントの一覧とその仕様を Markdown 形式でまとめてください
> エンドポイントごとに：メソッド、パス、リクエストボディ、レスポンス形式、認証の可否を含めてください
```

パターン 4: 変更履歴（CHANGELOG）を生成する

```
# git log を使って変更履歴を生成する
git log --oneline v1.0.0..HEAD | claude -p "このコミット一覧から CHANGELOG を生成してください。バグ修正、新機能、破壊的変更を分類して Markdown 形式でまとめてください"
```

19.6.3 Tips

つまづきポイント: ドキュメントは生成後に必ず確認が必要です。Claude はコードの意図を推測してドキュメントを書くため、実際の動作と異なる説明が含まれることがあります。特に「副作用」や「エラーハンドリング」の説明は要確認です。

ワンショットで素早くドキュメントを生成する場合：

```
cat src/utils/validation.ts | claude -p "このコードのドキュメントをコードに直接埋め込む形式で書いて。JSDoc 形式でコメントを付けたコード全体を出力して"
```


19.7 新規プロジェクトの立ち上げ

難易度: ★★☆☆ (中級)

19.7.1 いつ使うか

- 新しいプロジェクトをゼロから始めるとき
- 既存のプロジェクトに新しい機能モジュールを追加するとき
- ボイラープレートのセットアップを効率化したいとき

19.7.2 手順

フェーズ 1: CLAUDE.md の設計

新規プロジェクトでまず最初にすべきことは CLAUDE.md の作成です。プロジェクトの全体像を Claude に伝えることで、以後の作業精度が大幅に上がります。

```
mkdir my-new-project && cd my-new-project
claude
> 新しい Web アプリを作ります。以下の要件で CLAUDE.md を作成してください：
> - 概要：社内向けの勤怠管理システム
> - 技術スタック：Next.js 15, TypeScript, Prisma, PostgreSQL, Tailwind CSS
> - 開発規約：Conventional Commits, テストは Vitest
> - デプロイ先：Vercel
```

フェーズ 2: スキャフォールディング (雛形の生成)

```
claude
> 先ほど作成した CLAUDE.md の要件に基づいて、プロジェクトの雛形を作成してください
> 以下をセットアップしてください：
> 1. Next.js プロジェクトの初期化
> 2. TypeScript と ESLint の設定
> 3. Tailwind CSS の設定
> 4. Prisma の設定と初期スキーマ
> 5. Vitest のセットアップ
> 6. 基本的なディレクトリ構成の作成
```

フェーズ 3: 機能の実装

```
claude
> ユーザー認証機能を実装してください
> - メール/パスワードでのサインアップ・ログイン
> - JWT を使ったセッション管理
> - ログイン状態に応じたりダイレクト
> 既存の技術スタック (Next.js App Router, Prisma) に合わせて実装してください
```

19.7.3 Tips

つまづきポイント: 一度に全機能を実装しようとする、Claude のコンテキスト（覚えられる情報量）が足りなくなります。機能を小さな単位に分割して、1 機能ずつ実装するのが安全です。「ユーザー認証まで実装したら、次の会話セッションで商品管理機能を追加する」という進め方を心がけましょう。

大きなプロジェクトでのコンテキスト管理：

```
# セッションが長くなってきたらコンテキストを圧縮する
> /compact

# 新しいセッションで作業を再開するときは、まず現状把握から始める
claude
> @CLAUDE.md を読んで、このプロジェクトの構成を把握してください
> 次に src/ ディレクトリの構成を確認して、現在の実装状況を教えてください
```

19.8 非エンジニア向けワークフロー

プログラミングを知らなくても、Claude Code で自動化できる日常業務はたくさんあります。ここでは特に実用度の高いワークフローを 3 つ紹介します。

19.8.1 ファイル整理・一括リネーム

難易度: ★☆☆（初級）

いつ使うか: フォルダが散らかっているとき、写真や資料のファイル名をルールに沿って整理したいとき

```
claude
> デスクトップの「2025 年撮影」フォルダの中にある jpg ファイルを
> 「YYYYMMDD_連番.jpg」の形式（例：20250615_001.jpg）でリネームしてください
> ファイルの作成日時を日付として使ってください
> 実行前にどんな変更をするか一覧を見せてください
```

```
claude
> 「プロジェクト資料」フォルダの中のファイルを整理してください
> - PDF ファイル → 「PDF」サブフォルダに移動
> - Excel/CSV ファイル → 「データ」サブフォルダに移動
> - Word ファイル → 「文書」サブフォルダに移動
> - 画像ファイル (jpg/png) → 「画像」サブフォルダに移動
```

19.8.2 データ変換・集計レポート

難易度: ★☆☆（初級）

いつ使うか: CSV の集計、複数ファイルのデータ統合、Excel では難しいデータ加工をしたいとき

CSV ファイルを集計する

claude

- > デスクトップの「売上データ 2025.csv」を開いて、以下の集計を行ってください：
- > 1. 商品カテゴリ別の月次売上合計
- > 2. 売上上位 10 商品のランキング
- > 3. 前月比（対先月の売上変化率）
- > 結果を「売上集計レポート.csv」として保存してください

複数の CSV を 1 つにまとめる

claude

- > 「月次データ」フォルダの中にある CSV ファイルを全部読み込んで
- > 1 つのファイルにまとめてください
- > 各行にどの月のデータかわかるように「月」列を追加してください

つまづきポイント：「デスクトップの〇〇ファイル」と指示するとき、Claude Code は**現在のディレクトリ**を基準にファイルを探します。Claude Code を起動したディレクトリと、対象ファイルの場所が合っているか確認してください。cd ~/Desktop でデスクトップに移動してから Claude Code を起動するのが確実です。

19.8.3 テキスト変換・翻訳・要約

難易度: ★☆☆（初級）

いつ使うか: 大量のテキストファイルを一括処理したいとき、英語の資料を読みたいとき

フォルダ内の全テキストファイルを要約する

claude

- > 「会議メモ」フォルダの中にある .txt ファイルを全部読んで
- > それぞれ 3 行ずつ要点を抜き出し、「会議サマリー.md」にまとめてください
- > ファイル名と日付順に並べてください

英語の PDF を日本語に翻訳する

claude

- > デスクトップにある「report_en.pdf」を日本語に翻訳してください
- > 翻訳結果を「report_ja.txt」として保存してください
- > 専門用語はビジネス文書で使われる一般的な日本語訳を使ってください

テキストファイルを特定のフォーマットに変換する

claude

- > 「顧客リスト.txt」（名前とメールアドレスがタブ区切りで並んでいます）を
- > 「顧客リスト.csv」（カンマ区切り、ヘッダー行あり）に変換してください

19.8.4 ワンショットモードでの定期タスク自動化

定型作業はワンショットモード（-p）とシェルスクリプトを組み合わせで自動化できます：

```
#!/bin/bash
# 毎朝の日次レポート生成スクリプト例
# （このファイルを cron 等で毎朝実行する）

claude -p "昨日の売上データ ($(date -d yesterday +%Y%m%d)_sales.csv) を集計して、日次レ
ポートを daily_report_$(date +%Y%m%d).txt に保存してください"
```

Tips: `claude -p` に渡す指示文は、シングルクォートで囲むと `$` などの特殊文字が展開されません。変数を使いたい場合はダブルクォートを使いましょう。

19.9 次のステップ

各ワークフローを試してみたら、さらに効率を高めるための機能も探ってみてください。

エディタと組み合わせて使いたい方：

- [IDE 統合](#) (ページ 159) — VS Code や JetBrains と Claude Code を連携させる方法。エディタを離れずに AI の力を活用する

ターミナル操作の基礎から学びたい方：

- [非エンジニア向け Claude Code ガイド](#) (ページ 137) — ターミナルの基礎からわかりやすく解説

Claude Code のインストールがまだの方：

- [Claude Code をはじめよう](#) (ページ 129) — インストールから最初のコマンド実行まで

料金・プランを確認したい方：

- [Claude の料金プラン比較](#) (ページ 15) — Claude Code が使えるプランと利用量の目安

第20章 Claude Code の IDE 統合 (VS Code / JetBrains)

Claude Code はターミナルからも使えますが、普段の IDE から離れずに使える **IDE 統合**がとても便利です。VS Code と JetBrains 系 IDE (IntelliJ IDEA、WebStorm、PyCharm 等) 向けの拡張機能が提供されており、エディタのコンテキストを活かしながら AI アシスタントを活用できます。

この記事では、各 IDE への導入方法と、日常の開発フローへの組み込み方を解説します。

20.1 IDE 統合の概要

20.1.1 ターミナル版との違い

Claude Code はもともとターミナルで動作する CLI ツールですが、IDE 統合を使うと**エディタ内で完結**できるようになります。

	ターミナル版	IDE 統合
操作環境	ターミナル (コマンドライン)	VS Code / JetBrains のサイドパネル
コンテキスト	ファイルパスや内容を手動で指定	開いているファイル・選択中コードを自動認識
差分表示	ターミナル上のテキスト出力	インライン差分 (Diff ビュー) で視認しやすい
ファイル操作	コマンドで操作	エディタの変更として直接反映される
会話の継続	セッションが途切れることがある	ワークスペースに紐づいて会話が続く

20.1.2 IDE 統合のメリット

- **選択したコードをそのまま質問:** エディタで範囲選択した状態で Claude に投げると、そのコードが自動的にコンテキストに含まれる
- **インライン Diff 表示:** 変更提案が差分形式で表示され、Accept / Reject を選択できる
- **ファイルのシームレスな変更:** Claude が提案した変更がエディタに直接反映される。保存前に確認できる
- **統合ターミナル経由での実行:** IDE 内のターミナルで Claude Code が動くため、作業コンテキストが途切れない
- **CLI との会話共有:** VS Code 拡張機能は CLI を内包しており、CLI で開始した会話を IDE 内で継続できます。逆も同様です
- **チェックポイント機能:** 作業中に Git スナップショットを自動作成し、変更を安全にロールバックできます

20.1.3 Claude Code の 4 つの提供形態

Claude Code は以下の 4 つの形態で提供されています。どれも同じ AI 機能を利用でき、状況に応じて使い分けられます。

形態	特徴	向いている場面
CLI (ターミナル)	claude コマンドで起動	大規模な変更、スクリプトへの組み込み
IDE 拡張機能	VS Code / JetBrains のサイドパネル	日常のコーディング作業
デスクトップアプリ	Claude Desktop の Code モード	他の Claude 機能と併用したいとき
Web アプリ	claude.ai/code	ブラウザだけで使いたいとき

20.1.4 併用パターン

IDE 統合とターミナル版は排他的ではなく、状況に応じて使い分けるのが効果的です。

- **IDE 統合が向いている:** 特定のファイルを見ながらコードの修正・レビュー・質問をするとき
- **ターミナル版が向いている:** プロジェクト全体をまたぐ大規模な変更、CI/CD スクリプトへの組み込み、複数ステップの自動化タスク

20.2 VS Code 拡張機能

20.2.1 インストール方法

VS Code への Claude Code の導入は、VS Code の内蔵ターミナルを経由する形が標準です。拡張機能マーケットプレイスからインストールする方法と、CLI から直接起動する方法の 2 通りがあります。

動作要件: VS Code **1.98.0** 以上が必要です。また、**Cursor IDE** にも同一パッケージで対応しています。

拡張機能マーケットプレイスからインストール

1. VS Code の左サイドバーの「拡張機能」アイコン (または **Ctrl+Shift+X** / **Cmd+Shift+X**) をクリック
2. 検索欄に「Claude Code」と入力
3. Anthropic が公開している「Claude Code」拡張機能を選んで「インストール」をクリック

CLI から直接起動する方法

Claude Code がインストール済みであれば、任意のプロジェクトディレクトリで以下を実行するだけで VS Code と統合された状態で起動します。

```
claude
```

VS Code の統合ターミナルで実行すると、VS Code のコンテキスト (開いているファイル、ワークスペース情報) を自動的に認識します。

20.2.2 初期設定

拡張機能のインストール後、初回起動時に以下の設定が求められます。

1. **認証:** claude.ai のアカウントで認証（まだの場合は **Claude Code をはじめよう** (ページ 129) を先に読んでください)
2. **ワークスペースへのアクセス許可:** Claude Code がワークスペース内のファイルを読み書きする権限を付与する

つまづきポイント: 「Permission denied」が出る場合は、拡張機能の Permission 設定を確認してください。VS Code の設定 (Cmd+, / Ctrl+,) から「Claude Code」を検索し、Workspace Trust の設定を見直します。

20.2.3 サイドパネルでの会話

拡張機能をインストールすると、VS Code の左サイドバーに Claude Code のアイコンが追加されます。クリックするとサイドパネルが開き、そこでチャット形式で会話できます。

サイドパネルからは以下のような操作ができます。

- **自由形式の質問:** コードについての質問を自然言語で入力する
- **ファイルの参照:** @を入力するとファイル名補完が使える、特定のファイルをコンテキストに追加できる
- **選択範囲の送信:** エディタでコードを選択した状態でパネルに質問を入力すると、選択中のコードが自動的に含まれる

20.2.4 キーボードショートカット

Mac

操作	ショートカット
Claude Code パネルを開く / 閉じる	Cmd+Shift+C
選択したコードを Claude に送る	コード選択後 Cmd+Shift+Enter
インライン Diff を Accept	Cmd+Enter
インライン Diff を Reject	Cmd+Delete
会話をクリア	/clear をチャットに入力

Windows

操作	ショートカット
Claude Code パネルを開く / 閉じる	Ctrl+Shift+C
選択したコードを Claude に送る	コード選択後 Ctrl+Shift+Enter
インライン Diff を Accept	Ctrl+Enter
インライン Diff を Reject	Ctrl+Delete
会話をクリア	/clear をチャットに入力

Tips: キーバインドは VS Code の「キーボードショートカット」設定 (Cmd+K Cmd+S / Ctrl+K Ctrl+S) から自分好みに変更できます。「claude」で検索すると Claude Code 関連のショートカットが一覧表示されます。

20.2.5 インライン差分 (Diff) の操作

Claude Code が変更を提案すると、エディタ上に差分表示されます。削除行が赤、追加行が緑で表示されるいつもの Git Diff ビューとほぼ同じ見た目です。

- **Accept:** 変更を受け入れてファイルに反映する
- **Reject:** 変更を破棄して元のコードを維持する
- **Accept All / Reject All:** 複数の変更がある場合に一括操作する

つまづきポイント: インライン Diff が表示されず、チャットにコードブロックで返ってくる場合は、拡張機能が最新バージョンかどうか確認してください。古いバージョンではインライン Diff がサポートされていないことがあります。

20.3 JetBrains プラグイン

IntelliJ IDEA、WebStorm、PyCharm、GoLand など JetBrains 系の IDE でも Claude Code を使えます。

20.3.1 対応 IDE

- IntelliJ IDEA (Ultimate / Community)
- WebStorm
- PyCharm (Professional / Community)
- GoLand
- Rider
- CLion
- PhpStorm

20.3.2 インストール方法

JetBrains Marketplace からインストール

1. IDE のメニューから **Settings (設定)** を開く
 - macOS: Cmd+,
 - Windows / Linux: Ctrl+Alt+S
2. **Plugins (プラグイン)** を選択
3. **Marketplace** タブを開き、「Claude Code」を検索
4. Anthropic が公開している「Claude Code」プラグインの「Install」をクリック

5. IDE を再起動する

統合ターミナルから起動する方法

VS Code と同様に、JetBrains IDE の統合ターミナルで `claude` コマンドを実行すると、IDE のコンテキストを活用した状態で Claude Code が起動します。

```
claude
```

20.3.3 ツールウィンドウの使い方

プラグインをインストールすると、IDE の右サイドバーに「Claude Code」ツールウィンドウが追加されます（表示されない場合は **View > Tool Windows > Claude Code** から開く）。

機能的には VS Code 版と同等で、以下のことができます。

- チャット形式での質問・会話
- ファイルをコンテキストに追加（@ファイル名 形式）
- 選択したコードの送信
- インライン Diff の表示と Accept / Reject

20.3.4 キーボードショートカット

Mac

操作	ショートカット
Claude Code ツールウィンドウを開く / 閉じる	Cmd+Shift+C
選択したコードを Claude に送る	コード選択後 Cmd+Shift+Enter
インライン Diff を Accept	Cmd+Enter
インライン Diff を Reject	Cmd+Delete

Windows

操作	ショートカット
Claude Code ツールウィンドウを開く / 閉じる	Ctrl+Shift+C
選択したコードを Claude に送る	コード選択後 Ctrl+Shift+Enter
インライン Diff を Accept	Ctrl+Enter
インライン Diff を Reject	Ctrl+Delete

Tips: JetBrains IDE では **Settings > Keymap** から「Claude」で検索すると Claude Code 関連のキーバインドが一覧表示されます。競合しているショートカットは赤くハイライトされるので、その場で変更できます。

20.3.5 VS Code 版との機能差異

2026 年 3 月時点では、VS Code 版と JetBrains 版はほぼ同等の機能を提供しています。主な違いは以下の通りです。

機能	VS Code	JetBrains
チャットパネル	サイドバー	ツールウィンドウ
インライン Diff	あり	あり
ファイル参照 (@補完)	あり	あり
ターミナル統合	内蔵ターミナル経由	内蔵ターミナル経由
Git 統合	あり	あり
リファクタリング提案	あり	あり

つまづきポイント: JetBrains IDE では、プロジェクトを開いた状態でプラグインを有効にしないと、ツールウィンドウが表示されないことがあります。IDE を再起動するか、プロジェクトを閉じて開き直してみてください。

20.4 日常の開発フロー

IDE 統合を活用した実際の開発フローを紹介します。

20.4.1 コードを選択して質問する

最もよく使うパターンです。

1. エディタで気になるコード（関数・クラス・ファイル全体など）を選択する
2. Claude Code パネル / ツールウィンドウを開く
3. 質問を入力する（選択したコードは自動的にコンテキストに含まれる）

使用例:

このコードのパフォーマンス上の問題点を教えて

このクラスのメソッドが多すぎる。分割の方針を提案して

このコードに TypeScript の型を追加して

20.4.2 エラー箇所を修正依頼する

コンパイルエラーやランタイムエラーが出たとき。

1. エラーが出ているコードを選択する
2. エラーメッセージをコピーする
3. Claude Code パネルに「このエラーを修正して」とエラーメッセージを貼り付ける

使用例:

以下のエラーが出ています。修正してください。

```
Type 'string | undefined' is not assignable to type 'string'.  
  Type 'undefined' is not assignable to type 'string'.
```

Claude が修正案を提示したら、インライン Diff で確認して Accept / Reject を選択します。

20.4.3 テストを自動生成する

1. テストを書きたい関数・クラスを選択する
2. 「ユニットテストを書いて」と指示する

使用例:

選択したコードに対してユニットテストを書いて。
フレームワークは Vitest を使用。エッジケースも含めること。

テストファイルが提案されたら、新しいファイルとして保存するか、既存のテストファイルに追記するか選択できます。

20.4.4 コミットメッセージを生成する

1. IDE の統合ターミナルで `git diff --staged` の内容を確認する（またはそのまま Claude に渡す）
2. 「コミットメッセージを生成して」と依頼する

使用例（チャットパネル）:

```
@src/utils/date.ts @src/utils/date.test.ts  
これらの変更に対して Conventional Commits 形式のコミットメッセージを生成して
```

使用例（統合ターミナルで Claude Code を起動している場合）:

```
git diff --staged | claude -p "Conventional Commits 形式でコミットメッセージを生成して"
```

20.4.5 リファクタリングを依頼する

1. リファクタリング対象のコードを選択する（または @ファイル名 でファイル全体を参照する）
2. リファクタリングの方針を指示する

使用例:

```
@src/components/UserForm.tsx  
このコンポーネントが大きくなりすぎている。責務ごとに分割してほしい。  
ただし、props の型定義は変えないこと。
```

Tips: リファクタリングを依頼するときは「変えてほしくないこと」も一緒に伝えると、意図しない変更を防げます。例えば「テストは変更しないこと」「API の型定義は維持すること」など。

20.5 ターミナル版との使い分け

20.5.1 IDE 統合が向いている場面

- **ファイル単位の作業:** 特定のファイルを見ながらコードを書く・修正する・レビューする
- **インタラクティブな編集:** 変更案を確認しながら Accept / Reject を繰り返す作業
- **コンテキストが明確:** どのファイルのどの部分について作業するか決まっている
- **通常の開発作業:** フィーチャー実装、バグ修正、リファクタリングなど日常的な開発

20.5.2 ターミナル版が向いている場面

- **プロジェクト全体をまたぐ作業:** 複数ディレクトリにまたがるリファクタリング、一括検索・置換
- **自動化・スクリプト組み込み:** `claude -p` を使ったシェルスクリプトへの組み込み
- **大規模な変更:** 数十ファイル以上に影響する変更を一度に処理させるとき
- **CI/CD との連携:** GitHub Actions 等のパイプライン内での利用

組み合わせ例: IDE 統合で普段の実装を進めつつ、「全ファイルのコメントを英語に統一して」のような横断的な作業はターミナル版で処理する。

20.6 設定とカスタマイズ

20.6.1 VS Code の設定

VS Code の設定 (Cmd+, / Ctrl+,) で「Claude Code」を検索すると、以下の項目が設定できます。

設定項目	説明
<code>claudeCode.model</code>	使用するモデル (claude-sonnet-4-5 等) を指定する
<code>claudeCode.autoApprove</code>	自動承認するアクションの種類を設定する
<code>claudeCode.contextLines</code>	コンテキストとして送るコードの行数 (デフォルト: 50 行)

20.6.2 JetBrains の設定

Settings > Tools > Claude Code から以下を設定できます。

設定項目	説明
Model	使用するモデルを選択する
Auto-approve actions	自動承認するアクションの種類

次のページに続く

表 9 – 前のページからの続き

設定項目	説明
Context window	コンテキストに含める行数

20.6.3 キーバインドのカスタマイズ

Mac

VS Code: Cmd+K Cmd+S でキーボードショートカット設定を開き、「claude」で検索する。

JetBrains: Cmd+, で設定を開き、**Keymap** から「claude」で検索する。

Windows

VS Code: Ctrl+K Ctrl+S でキーボードショートカット設定を開き、「claude」で検索する。

JetBrains: Ctrl+Alt+S で設定を開き、**Keymap** から「claude」で検索する。

20.6.4 Permission（権限）設定

Claude Code が実行できる操作を制限・許可できます。チームで使う場合や、共有マシンで使う場合に特に重要です。

- **Read:** ファイルの読み取り
- **Write:** ファイルの書き込み・変更
- **Execute:** シェルコマンドの実行

各 IDE の Claude Code 設定から、どのアクションを自動承認するか・毎回確認するかを設定できます。

Tips: 初めて使うプロジェクトでは、Permission を厳しめ（自動承認なし）に設定しておき、Claude の動作に慣れてから緩めるのがおすすめです。

20.6.5 CLAUDE.md でプロジェクト固有の設定を追加する

CLAUDE.md にプロジェクト固有のルールを書くと、IDE 統合でも同様に読み込まれます。

IDE 利用時の注意

- コードの変更は必ずテストが通ることを確認してから **Accept** する
- 自動生成したコードはレビュー前にコミットしない
- リファクタリング時は型定義を変更しないこと

詳細は [Claude Code をはじめよう — CLAUDE.md の作成](#) (ページ 133) を参照してください。

20.7 トラブルシューティング

20.7.1 拡張機能 / プラグインが認識されない

症状: インストールしたはずなのに Claude Code のパネルが表示されない。

解決策:

1. IDE を完全に再起動する (ウィンドウを閉じてから再度開く)
2. 拡張機能が有効になっているか確認する (VS Code: 拡張機能一覧で「有効」と表示されているか)
3. Claude Code CLI 本体がインストールされているか確認する: `claude --version`

20.7.2 認証エラーが出る

症状: 「Authentication failed」や「Not authorized」というエラーが表示される。

解決策:

1. ターミナルで `claude` を実行して認証が通るか確認する
2. IDE を再起動して再度サインインする
3. API キーを使っている場合は環境変数が正しく設定されているか確認する

```
echo $ANTHROPIC_API_KEY
```

つまづきポイント: 環境変数を `.bashrc` や `.zshrc` に設定した場合、新しいターミナルセッションを開かないと反映されません。IDE を完全に再起動する必要があります。

20.7.3 インライン Diff が表示されない

症状: Claude の返答がチャットにテキストで表示されるだけで、エディタに差分が出ない。

解決策:

1. 拡張機能を最新バージョンに更新する
2. VS Code の場合: **Developer: Reload Window** コマンドを実行する (`Cmd+Shift+P` / `Ctrl+Shift+P` でコマンドパレットを開いて検索)
3. ファイルが保存されていない (`unsaved changes` がある) 場合に差分表示が抑制されることがあります。まず保存してから試す

20.7.4 レスポンスが遅い

症状: Claude の返答が遅い、タイムアウトする。

解決策:

1. コンテキストウィンドウ (送信するコードの量) を小さくする。設定の `contextLines` を減らす
2. 大きなファイル全体をコンテキストに含めるのではなく、関連する部分だけを選択して送る
3. `/compact` コマンドで会話を要約してトークン消費を減らす

20.7.5 VS Code で claude コマンドが見つからない

症状: 統合ターミナルで `claude` を実行すると「command not found」になる。

解決策: VS Code の統合ターミナルがシステムの PATH を正しく読み込んでいない可能性があります。

Mac

1. ターミナルアプリ (Terminal.app) で `which claude` を実行してインストール先を確認する
2. VS Code の **Settings > Terminal > Integrated: Inherit Env** が有効になっているか確認する
3. または、VS Code の統合ターミナルを `zsh -l` (ログインシェル) で起動するように設定する

```
// settings.json
{
  "terminal.integrated.defaultProfile.osx": "zsh",
  "terminal.integrated.profiles.osx": {
    "zsh": {
      "path": "zsh",
      "args": ["-l"]
    }
  }
}
```

Windows

1. WSL ターミナルで `which claude` を実行してインストール先を確認する
2. VS Code が WSL 拡張機能を通じて WSL に接続しているか確認する (左下のステータスバーに「WSL」と表示されているか)
3. WSL 拡張機能をインストールして、WSL 内のワークスペースで作業する

20.8 次のステップ

IDE 統合を使いこなせたら、Claude Code をさらに深く活用できる記事に進みましょう。

- [Claude Code 実践ワークフロー集](#) (ページ 147) — コードレビュー、バグ修正、テスト生成など実践的なワークフロー
- [Claude Code をはじめよう](#) (ページ 129) — ターミナル版の使い方、CLAUDE.md の設定方法
- [非エンジニア向け Claude Code ガイド](#) (ページ 137) — チームメンバーに Claude Code を紹介するときに共有できるガイド

第21章 Claude Desktop アプリの使い方

Claude Desktop は、Anthropic が提供するパソコン（macOS / Windows）向けのアプリです。ブラウザ版の claude.ai と同じ AI チャットが使えるだけでなく、パソコン内のファイルを直接操作したり、繰り返し作業を自動化したりできる「Cowork（コーワーク）」モードが搭載されています。

この記事では、Claude Desktop のインストールから各モードの使い方まで、順を追って解説します。

21.1 Claude Desktop とは — Web 版との何が違うの？

ブラウザで使える claude.ai（Web 版）と Claude Desktop アプリの最大の違いは、**パソコンとの連携の深さ**です。

機能	Web 版 (claude.ai)	Desktop アプリ
AI との会話	利用可	利用可
ファイルの添付	アップロードのみ	ローカルファイルに直接アクセス可
ドラッグ&ドロップ	利用可	利用可
Cowork（タスク自動化）	利用不可	利用可
スケジュール実行	利用不可	利用可
MCP 連携（外部ツール）	利用不可	利用可
通知	ブラウザ通知のみ	OS ネイティブ通知

21.1.1 Desktop アプリを使うメリット

1. ローカルファイルに直接アクセスできる

Web 版では、ファイルをいちいちアップロードする必要がありました。Desktop アプリでは、パソコン内のフォルダやファイルを指定して、Claude に直接読み書きさせることができます。「ドキュメントフォルダの資料を全部読んでまとめて」といった指示が一発で実行できます。

2. 繰り返し作業を自動化できる（Cowork モード）

「毎朝 9 時に昨日の売上データをまとめてレポートを作って」のように、定期的なタスクを自動実行させることができます。

3. 外部ツールと繋がられる（MCP 連携）

Slack、Google Drive、Notion など、普段使っているツールと Claude を連携させることで、複数のツールをまたいだ作業を一括して依頼できます。

21.1.2 Desktop アプリの 3 つのモード

Claude Desktop には以下の 3 つのモードがあります。

モード	概要	向いている用途
Chat	Web 版と同じ AI チャット	質問、文章作成、相談
Code	コード特化のチャット	プログラミング、スクリプト作成
Cowork	ファイル操作・自動化エージェント	ファイル整理、レポート作成、定期タスク

この記事では Chat と Cowork モードを中心に説明します。Code モードはエンジニア向けのため、ここでは割愛します。

21.2 インストール

Claude Desktop は claude.ai/download からダウンロードできます。

21.2.1 動作環境

- **macOS:** macOS 12 (Monterey) 以上
- **Windows:** Windows 10 / 11 (64bit)

21.2.2 インストール手順

Mac

1. claude.ai/download にアクセスします
2. 「Download for macOS」ボタンをクリックして **.dmg** ファイルをダウンロードします
3. ダウンロードされた **.dmg** ファイルをダブルクリックして開きます
4. 表示されたウィンドウで、Claude のアイコンを「アプリケーション」フォルダにドラッグします
5. 「アプリケーション」フォルダから Claude を起動します
6. 初回起動時に「インターネットからダウンロードされたアプリケーションです」という確認ダイアログが表示される場合は「開く」をクリックします

Windows

1. claude.ai/download にアクセスします
2. 「Download for Windows」ボタンをクリックして **.exe** インストーラーをダウンロードします
3. ダウンロードされた **.exe** ファイルをダブルクリックしてインストーラーを起動します
4. 「このアプリがデバイスに変更を加えることを許可しますか？」というダイアログが表示されたら「はい」をクリックします
5. 画面の指示に従ってインストールを完了します

6. スタートメニューまたはデスクトップのショートカットから Claude を起動します

つまづきポイント: macOS で「壊れているため開けません」と表示される場合は、「システム設定 > プライバシーとセキュリティ」を開き、「このまま開く」をクリックして許可してください。これは Apple が認証していないアプリを保護する機能（Gatekeeper）によるものです。

21.2.3 ログインとアカウント設定

インストールが完了したら、Claude Desktop を起動してログインします。

1. 起動するとログイン画面が表示されます
2. claude.ai で使っているメールアドレスとパスワードを入力します
3. Google / Apple アカウントでのログインも可能です
4. ログインが完了すると、Claude Desktop のメイン画面が表示されます

まだアカウントを持っていない場合は、claude.ai から無料で作成できます。

21.3 Chat モード — Web 版と同じ感覚で使える

Chat モードは、ブラウザ版の claude.ai と同じ AI チャット機能です。インストール後すぐに使い始められます。

21.3.1 基本の使い方

1. 左側のサイドバーで「**Chat**」タブを選択します（デフォルトで Chat モードになっています）
2. 画面下部のテキスト入力欄にメッセージを入力します
3. Enter キー（または送信ボタン）を押して送信します

Web 版と同じように、文章作成、翻訳、質問応答、アイデア出しなど、さまざまな用途に使えます。

21.3.2 Desktop アプリならではの便利機能

ローカルファイルのドラッグ&ドロップ

パソコン内のファイルを、チャット画面に直接ドラッグ&ドロップするだけで添付できます。Web 版のようにブラウザのアップロード画面を開く手間が不要です。

対応しているファイル形式:

- PDF ドキュメント
- Word ファイル (.docx)
- Excel ファイル (.xlsx)
- テキストファイル (.txt、.csv など)
- 画像ファイル (.png、.jpg など)

OS ネイティブの通知

Claude が長い回答を生成し終わると、パソコンの通知センターに通知が届きます。長い文章の作成を依頼して、その間に別の作業をしても、完了を見逃しません。

キーボードショートカット

Mac

操作	ショートカット
新しい会話を開始	Cmd + N
アプリを前面に表示	Cmd + Tab で切り替え
設定を開く	Cmd + ,

Windows

操作	ショートカット
新しい会話を開始	Ctrl + N
設定を開く	Ctrl + ,

つまづきポイント: Desktop アプリでの Chat は Web 版と会話履歴が共有されています。Web で始めた会話を Desktop で続けることも、その逆も可能です。

21.4 Cowork モード — Claude にタスクを丸ごと任せる

Cowork（コーワーク）は、Claude Desktop アプリの最も強力な機能です。ファイルの読み書き、外部サービスとの連携、繰り返し作業の自動化など、AI に実際の「仕事」を任せることができます。

21.4.1 Chat との違い

Chat モードでは Claude は「質問に答える」役割ですが、Cowork モードでは Claude が自分から動いて作業を進める「エージェント」として動きます。

Chat モードの例:

あなた: 「会議のアジェンダを作って」
Claude: 「はい、以下のアジェンダ案はいかがでしょうか… (テキストを返す)」

Cowork モードの例:

あなた: 「今週の議事録フォルダのファイルを全部読んで、
アクションアイテムの一覧を Excel に作って保存して」
Claude: (フォルダを開く → 各ファイルを読む → Excel を作成 → 保存する を自動実行)

21.4.2 Cowork が使えるプラン



警告:

Cowork は **Pro プラン以上**で利用できます。Free プランでは Chat モードと Code モードのみ使用可能です。まずは無料プランで Claude の使い心地を試してから、アップグレードをご検討ください。

プラン	月額	Cowork
Free	無料	利用不可
Pro	\$20	利用可
Max 5x	\$100	利用可（高い利用上限）
Max 20x	\$200	利用可（最優先アクセス）

21.4.3 Cowork の初期設定

Cowork モードを初めて使う際は、Claude がアクセスできるフォルダを設定します。

1. 左サイドバーの「**Cowork**」タブをクリックします
2. 「フォルダを追加」をクリックします
3. Claude にアクセスを許可するフォルダを選択します（例: 「ドキュメント」フォルダ）
4. 「許可」をクリックして確定します



注釈:

許可したフォルダ内のファイルだけに Claude がアクセスできます。許可していないフォルダのファイルは読めません。最初は「ドキュメント」フォルダだけ許可して、慣れたら範囲を広げていくのがおすすめです。

21.4.4 Cowork の活用例

例 1: 大量のファイルを一括処理する

「先月の請求書フォルダにある PDF をすべて読んで、取引先名・金額・日付を一覧にした Excel ファイルを作って」のように、大量のファイルをまとめて処理できます。手作業では数時間かかる作業が、数分で完了します。

例 2: 議事録のサマリーを自動作成する

「議事録フォルダの今週分の Word ファイルを読んで、各会議の決定事項とアクションアイテムを箇条書きでまとめた週次サマリーを作成して、デスクトップに保存して」

例 3: 定期レポートの自動作成（スケジュール機能）

「毎朝 9 時に、売上データフォルダの最新 CSV を読んで、前日比を計算した日次レポートを作成して、reports フォルダに保存して」のように、定期的なタスクを自動化できます。

つまづきポイント: Cowork のスケジュールタスクは、パソコンの電源が入っており Claude Desktop が起動している場合のみ実行されます。スリープ中や電源オフのときはスキップされます。

Cowork の詳細な使い方（スケジュール設定、外部サービス連携など）は「[Cowork 入門 — Claude にタスクを任せよう](#) (ページ 179)」を参照してください。

21.5 MCP 連携 — 外部ツールと繋げる

MCP (Model Context Protocol) は、Claude と外部ツールを連携させる仕組みです。この機能を使うと、Slack、Google Drive、Notion などの普段使っているサービスに Claude がアクセスできるようになります。

21.5.1 MCP でできること

MCP を設定すると、たとえば以下のようなことが可能になります。

- 「Slack の #general チャンネルの今日のメッセージを要約して」
- 「Google Drive の特定フォルダにある資料を読んで分析して」
- 「Notion のデータベースに今日の作業ログを追加して」

21.5.2 主な MCP サーバー

ツール	できること
Google Drive	ファイルの読み込み、Drive 内の検索
Slack	メッセージの読み取り、チャンネルの要約
Notion	ページの読み書き、データベースの操作
GitHub	リポジトリの参照、Issue の管理
Brave Search	Web 検索の結果を Claude に取り込む



警告:

MCP の設定はやや技術的な操作（設定ファイルの編集など）が必要です。このページでは概要のみ紹介しています。詳しい設定方法は「[MCP 連携ガイド](#) (ページ 345)」を参照してください。

21.5.3 MCP の設定場所（概要）

Claude Desktop の設定ファイル (claude_desktop_config.json) に MCP サーバーの情報を記述することで連携できます。設定ファイルの場所は以下の通りです。

Mac

~/Library/Application Support/Claude/claude_desktop_config.json

Windows

```
%APPDATA%\Claude\claude_desktop_config.json
```

MCP の詳細な設定手順については、別途ガイドページを用意しています。

21.6 Chrome 拡張・Office アドインとの連携

Claude Desktop のほか、以下のツールからも Claude を利用できます（詳細は将来の別ガイドで解説予定）。

Chrome 拡張

Chrome ブラウザに Claude の拡張機能をインストールすると、閲覧中のウェブページの内容を Claude に直接渡して質問できるようになります。ウェブ上の記事や PDF の要約、調査作業がスムーズになります。

Excel / PowerPoint アドイン

Microsoft Excel や PowerPoint に Claude のアドインを追加すると、Office アプリ内から直接 Claude に指示を出せます。「このスプレッドシートのデータを分析して」「このスライドの文章を改善して」といった作業を、アプリを切り替えることなく実行できます。



注釈:

Chrome 拡張や Office アドインの詳細な設定方法は、今後のガイドページで詳しく解説します。

21.7 よくある質問

Q. Web 版（claude.ai）とデータは共有されますか？

はい。Chat の会話履歴はクラウドに保存され、Web 版と Desktop アプリで共有されます。どちらからでも過去の会話にアクセスできます。

Q. Desktop アプリは無料で使えますか？

Chat モードと Code モードは Free プランでも利用できます。Cowork モードを使うには Pro プラン以上が必要です。

Q. アプリを常時起動しておく必要がありますか？

Chat モードは必要なときだけ起動すれば OK です。スケジュール機能（Cowork）を使う場合は、タスクの実行時刻にアプリが起動している必要があります。

つまづきポイント: Desktop アプリのアップデートは自動では行われない場合があります。アプリメニューの「ヘルプ > アップデートを確認」（macOS）または「Claude についての情報」（Windows）から定期的にアップデートを確認してください。最新バージョンを使うことで、新機能や不具合の修正が適用されます。

21.8 次のステップ

Claude Desktop の基本を理解できたら、さらにステップアップしましょう。

- [Cowork 入門 — Claude にタスクを任せよう](#) (ページ 179) — Cowork の詳細な使い方 (スケジュール、外部サービス連携) を学ぶ
- [Claude Web \(claude.ai\) の使い方](#) (ページ 89) — ブラウザ版との使い分け方を知る
- [プロンプトの基本](#) (ページ 239) — Claude にうまく指示を出すテクニックを学ぶ
- [料金プランの比較](#) (ページ 15) — 自分に合ったプランを選ぶ

第22章 Cowork 入門 — Claude にタスクを任せよう

Claude の **Cowork**（コーワーク）は、Claude Desktop アプリに搭載されたナレッジワーク向けのエージェント機能です。普通のチャットとは異なり、ファイルの読み書き、外部サービスとの連携、繰り返し作業の自動化など、実際の「仕事」を Claude に任せられます。

この記事では、Cowork のセットアップから最初のタスク実行まで、ステップバイステップで解説します。

22.1 Cowork とは — 「会話」ではなく「タスク実行」

普通のチャット（Chat モード）では、Claude は「質問に答える」「文章を書いてもらう」など、返事を返すことが中心です。一方、Cowork は Claude が自分から動いて作業を進めるモードです。

たとえば、こんなことが可能です。

- **ファイル操作**: 「このフォルダの PDF をすべて読んで、内容を一覧にして」と指示すると、Claude がフォルダ内のファイルの一つひとつ開いて情報をまとめてくれます
- **スケジュール実行**: 「毎朝 9 時に今日のタスクをまとめてメールに送って」のように、定期的なタスクを自動化できます
- **外部サービス連携**: Slack や Notion、Gmail などと連携し、「Slack の特定チャンネルをサマリーして Notion のページに書き込んで」といった複数ツールをまたぐ作業もこなします

22.1.1 Chat モードとの違い

	Chat モード	Cowork モード
動かし方	質問・返答のやりとり	タスクを指示して実行させる
ファイルアクセス	添付ファイルのみ	パソコン内のフォルダ・ファイルに直接アクセス可
外部サービス	なし	Slack、Notion、Gmail 等と連携可
定期実行	なし	スケジュール設定で自動化可
向いている用途	質問、相談、文章作成	ファイル整理、レポート作成、反復作業の自動化

Cowork はどんな人に向いている？: 毎日同じような作業を繰り返している方、大量のファイルをまとめて処理したい方、複数のツールをまたいで情報を整理したい方に特に役立ちます。プログラミングの知識がなくても使えます。

Microsoft 365 でも Cowork 技術が使える: 2026 年 3 月、Microsoft は「**Copilot Cowork**」を発表しました。Anthropic の Claude Cowork 技術基盤を採用しており、Microsoft 365 環境内で自

律的にタスクを実行できます。会議スケジュールの調整からメール調査、PowerPoint 資料の作成までを一連の流れで自動処理します。現在は早期プログラム「Frontier」向けに提供中です。

22.1.2 利用できるプラン

Cowork は **Pro プラン**以上で利用できます。Free プランでは Chat モードと Code モードのみ使用可能です。まずは無料プランで Claude の使い心地を確認してから、必要に応じてアップグレードするのがおすすめです。

プラン	月額	Cowork
Free	無料	利用不可
Pro	\$20	利用可（通常制限あり）
Max 5x	\$100	利用可（高い上限）
Max 20x	\$200	利用可（最優先）

22.2 動作環境 — デスクトップアプリが必要

Cowork は **Claude Desktop アプリ**（パソコン用のアプリ）の機能です。ブラウザ版の `claude.ai` では使えません。

22.2.1 対応 OS

- **macOS:** macOS 12（Monterey）以上
- **Windows:** Windows 10 / 11（64bit）

22.2.2 なぜデスクトップアプリなのか

Cowork がパソコン上のファイル进行操作したり、外部サービスと連携したりするためには、ブラウザではできないシステムレベルのアクセスが必要です。Claude Desktop アプリをインストールすることで、Claude が安全な作業スペース（パソコン内の隔離された専用環境）を通じてこれらの操作を実行できるようになります。

プライバシーと安全性: Claude がアクセスできるフォルダやサービスは、あなたが明示的に許可したものだけです。初回のフォルダアクセス時には確認ダイアログが表示されます。

22.3 セットアップ — 3 ステップで準備完了

22.3.1 ステップ 1: Claude Desktop アプリをインストールする

Mac

1. claude.ai/download にアクセスします
2. 「Download for macOS」ボタンをクリックして `.dmg` ファイルをダウンロードします

3. ダウンロードした **.dmg** ファイルを開き、Claude アイコンを「アプリケーション」フォルダにドラッグします
4. アプリケーションフォルダから Claude を起動します

Windows

1. claude.ai/download にアクセスします
2. 「Download for Windows」ボタンをクリックして **.exe** インストーラーをダウンロードします
3. ダウンロードした **.exe** ファイルをダブルクリックしてインストーラーを起動します
4. 画面の指示に従ってインストールを完了します
5. スタートメニューまたはデスクトップのショートカットから Claude を起動します

22.3.2 ステップ 2: ログインして Cowork モードに切り替える

1. Claude Desktop を起動すると、ログイン画面が表示されます
2. claude.ai のアカウント（メールアドレス）でログインします
3. アプリが起動したら、左側のサイドバー（または画面上部のメニュー）から「**Cowork**」タブをクリックします

つまづきポイント: ログイン後に Cowork タブが見当たらない場合は、Pro プラン以上のアカウントでログインしているか確認してください。Free プランでは Cowork タブが表示されません。

22.3.3 ステップ 3: フォルダアクセスを許可する

Cowork を初めて起動すると、「どのフォルダにアクセスしますか？」という設定画面が表示されます。

1. 「**フォルダを追加**」をクリックします
2. Claude にアクセスを許可したいフォルダを選択します（例: 「ドキュメント」フォルダや仕事用のフォルダ）
3. 「許可」をクリックして確定します

許可したフォルダ内のファイルにだけ、Claude がアクセスできるようになります。許可していないフォルダのファイルは読めません。これはセキュリティ上、とても大切な設計です。

おすすめの初期設定: 最初は「デスクトップ」や「ドキュメント」フォルダだけを許可して始めましょう。後から追加・削除できるので、試しながら範囲を広げていくのが安全です。

22.4 最初のタスクを実行する

セットアップが完了したら、実際にタスクを実行してみましょう。

22.4.1 やってみよう: フォルダのファイル一覧を作る

一番かんたんな入門タスクとして、フォルダ内のファイル名を一覧にしてみましょう。

入力例:

デスクトップの「プロジェクト資料」フォルダの中にあるファイルをすべて確認して、ファイル名と更新日時の一覧を表で作成してください。

Claude の動き:

1. 「プロジェクト資料」フォルダを開きます
2. ファイルを一つひとつ確認します
3. 表形式で一覧を作成してチャット画面に表示します

タスクが実行されると、画面右側に「Claude が実行した操作の記録」(操作ログ)が表示されます。Claude が何をしたかをリアルタイムで確認できます。

22.4.2 やってみよう: 複数ファイルをまとめて処理する

次は少し応用。複数の PDF やドキュメントの内容をまとめてみましょう。

入力例:

「議事録」フォルダの中にある Word ファイルをすべて読んで、それぞれの会議の日付・参加者・決定事項をまとめた一覧表を作ってください。まとめた内容は「議事録サマリー.docx」というファイル名でデスクトップに保存してください。

この例では Claude が以下の手順を自動で実行します。

1. 「議事録」フォルダ内の全 Word ファイルを開く
2. 各ファイルから日付・参加者・決定事項を抽出する
3. 表形式にまとめる
4. デスクトップに 議事録サマリー.docx として保存する

つまづきポイント: Claude が「ファイルが見つかりません」と答えた場合、そのフォルダがアクセス許可に含まれているか確認してください。設定 > フォルダアクセスで確認・追加できます。

22.5 タスクの書き方 — 良い指示の出し方

Cowork の結果は、**タスクの指示をどう書くか**で大きく変わります。具体的であるほど、期待通りの結果が返ってきます。

22.5.1 悪い例と良い例

例 1: ファイル整理

指示の内容	
悪い例	「ファイルを整理して」
良い例	「デスクトップにある 2025 年以前の日付のファイルを「アーカイブ」フォルダに移動して。フォルダがなければ作成して」

例 2: レポート作成

指示の内容	
悪い例	「売上データを分析して」
良い例	「ドキュメントフォルダの「sales-2026Q1.csv」を読んで、月別の売上合計と前月比 (%) を表にまとめて。グラフも作ってレポートに含めて」

例 3: 定期タスク

指示の内容	
悪い例	「毎日レポートを送って」
良い例	「毎朝 8:30 に「売上データ」フォルダの最新ファイルを確認して、前日との差分をまとめた日次レポートを sales@example.com に送って」

22.5.2 タスク指示のコツ

- **対象ファイルやフォルダのパスを明確に:** 「このファイル」ではなく「デスクトップの〇〇フォルダの〇〇.xlsx」のように
- **完成形を伝える:** 「整理して」より「〇〇というフォルダに日付順で保存して」
- **保存場所を指定する:** 「保存して」だけでなく「デスクトップに〇〇.docx として保存して」
- **条件を明確に:** 「古いファイル」ではなく「2024 年より前のファイル」

22.6 進行状況の確認と方向転換

Cowork がタスクを実行している間、何が起きているか確認できます。

22.6.1 実行ログを見る

タスク実行中、画面右側（または下部）に「操作ログ」が表示されます。

- どのファイルを開いたか
- 何を読み取ったか

- どんな操作をしたか

が順番に記録されます。「今どこまで進んでいるか」がわかるので安心して待てます。

22.6.2 途中で止める

タスクが期待と違う方向に進んでいると感じたら、いつでも止められます。

- 「**停止**」ボタン（または「Stop」ボタン）をクリックすると、Claude はすぐに作業を止めます
- その後、追加で指示を入力して方向を修正できます

修正の例:

ちょっと待って。アーカイブフォルダに移動するのはいいけど、
2024 年のファイルはそのままにして。2023 年以前だけ移動してください。

このように、途中で指示を追加・修正しながらタスクを進められるのが Cowork の便利な点です。

つまづきポイント: Claude が止まらない場合は、アプリウィンドウを閉じてもタスクは停止します。重要なファイルを扱う前は、必ずバックアップを取る習慣をつけましょう。

22.7 スケジュールタスク — 繰り返しの仕事を自動化する

毎日・毎週同じ作業を繰り返している場合、スケジュール機能を使えば Claude に自動化してもらえます。

22.7.1 スケジュールの設定方法

1. Cowork のタスク入力欄に「毎日〇時に～してください」と入力します
2. Claude がスケジュール設定を提案します
3. 確認して「設定する」をクリックします
4. 設定画面（Cowork > スケジュール）でいつでも確認・変更・削除できます

22.7.2 スケジュールタスクの活用例

日次レポートの自動作成:

毎朝 8 時に、「報告書」フォルダの最新ファイルを読んで、
前日分のサマリー（箇条書き 5 項目）を作成して、
「日次サマリー.txt」というファイルをデスクトップに保存してください。

週次サマリーの送付:

毎週金曜日の 17 時に、今週追加・更新されたファイルの一覧をまとめて、
team@example.com に週次サマリーとして送ってください。
件名は「週次ファイル更新サマリー（〇月〇日週）」にしてください。

定期的なデータ整理:

毎月 1 日の 9 時に、「受信ファイル」フォルダの中身を確認して、前月分のファイルを「アーカイブ/〇年〇月」フォルダに移動してください。

つまづきポイント: スケジュールタスクはパソコンの電源が入っており、Claude Desktop が起動している場合のみ実行されます。電源が切れていたり、スリープしていたりすると、その時間のタスクはスキップされます。重要な自動化はタスクが実行されたかを翌日に確認する習慣をつけましょう。

22.8 外部サービス連携 — Slack・Notion・Gmail と繋げる

Cowork はプラグインを使って外部サービスと連携できます。連携することで、複数のツールをまたぐ作業も Claude 一つでこなせるようになります。

22.8.1 連携できる主なサービス

- **Slack:** チャンネルのメッセージを読んでサマリーを作成、重要な情報を Notion などに転記
- **Notion:** ページの作成・更新、データベースへの書き込み
- **Gmail / Google Workspace:** メールの送受信、カレンダーの確認・作成、スプレッドシートの読み書き
- **GitHub:** コードのレビュー、イシューの管理（エンジニア向け）
- **その他:** Jira、Asana、Dropbox など

22.8.2 プラグインの追加方法

1. Cowork の画面左側（または設定メニュー）から「**プラグイン**」をクリックします
2. 連携したいサービスのプラグインを選択します
3. 「インストール」をクリックし、そのサービスの認証（ログイン）を行います
4. 権限の確認画面が表示されるので、必要な権限だけを許可して「接続」をクリックします

Slack + Notion 連携の活用例:

Slack の #general チャンネルの今日のメッセージをすべて読んで、重要な情報・決定事項・アクションアイテムをまとめて、Notion の「ミーティングメモ」データベースに今日の日付で追加してください。

つまづきポイント: プラグインを追加したのに「そのサービスにアクセスできません」と表示される場合は、認証が切れている可能性があります。プラグイン設定画面から「再接続」をクリックして、もう一度ログインしてみてください。

プラグインの詳細な使い方については「[Cowork プラグイン — 機能を拡張しよう](#) (ページ 187)」を参照してください。

22.9 セキュリティと安全性

「Claude がパソコンの中を自由に操作できるなんて怖い」と感じる方もいるかもしれません。Cowork にはいくつかの安全の仕組みが組み込まれています。

22.9.1 許可したフォルダだけアクセスできる

セットアップ時に許可したフォルダ以外には、Claude はアクセスできません。フォルダの追加・削除は「設定 > フォルダアクセス」からいつでも変更できます。

22.9.2 削除は確認が必要

ファイルの削除など、元に戻せない操作をしようとした場合、Claude は必ず確認を求めます。「本当に削除しますか？」という確認なしに、ファイルが消えることはありません。

22.9.3 操作ログがすべて残る

Claude が実行した操作はすべてログに記録されます。「Claude が何をしたか」を後から確認できるので、問題が起きたときに原因を特定しやすくなっています。

22.9.4 ネットワーク通信の制御

Cowork がインターネットに接続する場合（外部サービス連携など）、最初に許可を求めます。あなたが許可した範囲以外の外部通信は行いません。

おすすめのセキュリティ習慣: 重要なファイルを扱う前は、あらかじめバックアップを取っておきましょう。また、見知らぬプラグインは信頼できるものだけをインストールし、不要になったプラグインはこまめに削除する習慣をつけてください。

22.10 次のステップ

Cowork の基本をマスターしたら、さらに活用の幅を広げましょう。

- [Cowork プラグイン — 機能を拡張しよう](#) (ページ 187) — Slack、Notion、Gmail などの外部サービスとの連携を深める
- [Claude Desktop アプリの使い方](#) (ページ 171) — Chat モードと Cowork モードを使い分けるコツ
- [Claude Web \(claude.ai\) の使い方](#) (ページ 89) — ブラウザ版との違いと使い分け方
- [プロンプトの基本](#) (ページ 239) — Claude にうまく指示を出すテクニックを学ぶ

第23章 Cowork プラグイン — 機能を拡張しよう

Claude の **Cowork** は、プラグインを追加することで使えるサービスの範囲を大幅に広がられます。標準状態ではパソコン上のファイル操作が中心ですが、Slack・Notion・Google Drive・GitHub などのプラグインを追加すると、複数のツールをまたいだ作業を Claude ひとつでこなせるようになります。

この記事では、プラグインの概念から導入・管理の手順、主要プラグインの活用法までを解説します。プログラミングの知識がなくても、公式プラグインであれば誰でも数分でセットアップできます。

23.1 プラグインとは — Cowork の「連携ブリッジ」

プラグインは、Cowork と外部サービスの間を繋ぐ「連携ブリッジ」です。プラグインを追加することで、Claude が外部サービスにアクセスして情報を読み書きできるようになります。

23.1.1 何ができるようになるか

プラグインなしの状態では、Claude が操作できるのは「許可したフォルダ内のファイル」だけです。プラグインを追加すると、次のようなことが可能になります。

プラグイン	できること
Slack	チャンネルのメッセージを読む、メッセージを送る
Notion	ページを作成・更新する、データベースに書き込む
Google Drive	ファイルを検索・読み込む、スプレッドシートを編集する
GitHub	リポジトリを読む、イシューを作成・更新する
Gmail	メールを読む、メールを送信する

これらを組み合わせると、たとえば「Slack のメッセージを読んで Notion に転記する」「Google Drive のデータを分析してメールでレポートを送る」といった、複数ツールをまたぐ自動化が実現します。

23.1.2 MCP サーバーとの関係（技術的な補足）

プラグインは、内部的には **MCP (Model Context Protocol) サーバー** という仕組みの上に構築されています。MCP はツールと AI を繋ぐ標準的な通信規格で、公式プラグインはその規格に沿って作られた「使いやすいパッケージ」です。

非エンジニアの方は「プラグインを追加すると Claude が外部サービスを使えるようになる」と理解すれば十分です。技術的な詳細に興味がある方は「[MCP ガイド — Claude に外部ツールをつなげる](#) (ページ 345)」を参照してください。

23.2 公式プラグインの導入手順

公式プラグインはプラグインストアから数ステップでインストールできます。

23.2.1 ステップ 1: プラグインストアを開く

1. Claude Desktop アプリを起動します
2. 左サイドバーの「Cowork」タブを選択します
3. 画面左下（または設定メニュー）にある「プラグイン」アイコンをクリックします
4. プラグインストア（インストール可能なプラグインの一覧）が表示されます

23.2.2 ステップ 2: プラグインを選んでインストール

1. 導入したいプラグインを選択します
2. プラグインの詳細画面が開きます。「必要な権限」を確認します
3. 「インストール」ボタンをクリックします

23.2.3 ステップ 3: 外部サービスで認証する

インストール後、外部サービスへのログイン（認証）が必要になります。

1. 「接続する」または「認証する」ボタンをクリックします
2. ブラウザが開き、対象サービスのログイン画面（Slack、Notion など）が表示されます
3. そのサービスのアカウントでログインします
4. 「Claude に以下の権限を許可しますか？」という確認画面が表示されます
5. 内容を確認し、「許可する」をクリックします
6. Claude Desktop に戻り、接続が完了したことを確認します

つまづきポイント: 認証後に「接続できませんでした」と表示される場合は、ブラウザのポップアップブロックが原因のことがあります。ブラウザの設定でポップアップを許可するか、表示された URL を手動でコピーしてブラウザに貼り付けて認証を完了させてください。

23.2.4 ステップ 4: 動作確認

インストールが完了したら、実際に動くか確認しましょう。

Slack プラグインの確認例:

Slack の #general チャンネルの最新のメッセージを 5 件読んで、内容を教えてください。

Claude が Slack のメッセージを返してきたら、接続は成功です。

23.3 プラグインの管理

23.3.1 プラグインの一覧を確認する

インストール済みのプラグインは「設定 > プラグイン」から一覧で確認できます。各プラグインについて以下の情報が表示されます。

- **接続状態:** 有効（緑） / 無効（グレー） / 要再認証（赤・黄）
- **最終更新日:** プラグインのバージョン情報
- **付与済み権限:** そのプラグインが持っているアクセス権限の内容

23.3.2 プラグインを一時的に無効にする

一時的に使わなくなったプラグインは、無効化することで権限を保持したまま停止できます。

1. 「設定 > プラグイン」でプラグイン一覧を開きます
2. 対象のプラグインのトグルスイッチをオフにします
3. 「無効化しますか？」の確認が出たら「無効化」をクリックします

無効化したプラグインは、後から同じ手順でトグルをオンにするだけで再び使えます。再認証は不要です。

23.3.3 プラグインを削除する

プラグインを完全に削除すると、アクセス権限も削除されます。外部サービスとの連携を解除したい場合に使います。

1. 「設定 > プラグイン」でプラグイン一覧を開きます
2. 対象のプラグインを選択し、「削除」をクリックします
3. 「本当に削除しますか？」の確認が出たら「削除」をクリックします

つまづきポイント: プラグインを削除しても、外部サービス側（Slack、Notion など）の認証は残ったままになることがあります。完全に連携を解除したい場合は、そのサービスのアカウント設定画面から Claude の連携を削除する操作も別途必要です。

23.3.4 アップデートの確認

プラグインに新しいバージョンがあると、プラグイン一覧に「更新可能」のバッジが表示されます。

1. 「設定 > プラグイン」でプラグイン一覧を開きます
2. 「更新可能」と表示されているプラグインを選択します
3. 「更新する」をクリックします

アップデート後に新しい権限が追加されている場合は、再度確認画面が表示されます。

23.4 主要プラグイン紹介

よく使われる公式プラグインを紹介します。

23.4.1 Slack — チームのコミュニケーションを自動化

できること:

- 指定チャンネルのメッセージを読んでサマリーを作成
- 特定のキーワードを含むメッセージを検索
- メッセージを送信・投稿
- スレッドの内容を抽出して資料を作成

活用例:

今日の #sales チャンネルの全メッセージを読んで、重要な情報・決定事項・アクションアイテムを箇条書きでまとめてください。

先週一週間の #general チャンネルの内容をサマリーして、「週次ハイライト.txt」としてデスクトップに保存してください。

必要な権限: チャンネルの読み取り、メッセージの送信（送信機能を使う場合）

つまづきポイント: プライベートチャンネルのメッセージは、そのチャンネルのメンバーとして認証しているアカウントでのみ読み取れます。招待されていないプライベートチャンネルにはアクセスできません。

23.4.2 Notion — ナレッジベースと連携

できること:

- ページの作成・更新
- データベースへのレコード追加・更新
- 既存ページの内容を読み取り
- ページ一覧の検索

活用例:

「ミーティングメモ」データベースに今日の日付で新しいページを作成して、以下の内容を書き込んでください:

- 日付: 2026 年 3 月 24 日
- 参加者: 田中、鈴木、山田
- 決定事項: Q2 の目標を〇〇に設定

Notion の「プロジェクト管理」データベースでステータスが「進行中」のタスクを一覧にして教えてください。

必要な権限: ページの読み取り、ページの作成・編集

23.4.3 Google Drive / Google Workspace — G Suite と連携

できること:

- Google ドライブのファイル検索・読み込み
- Google スプレッドシートのデータ読み書き
- Google ドキュメントの内容読み取り・編集
- Google カレンダーのイベント確認・作成

活用例:

Google スプレッドシートの「売上管理 2026.xlsx」を読んで、先月の売上合計と前月比を教えてください。

今週の Google カレンダーの予定を確認して、会議が多い日と空いている時間帯を教えてください。

必要な権限: Google ドライブのファイル読み取り（書き込みが必要な場合は書き込み権限も）

つまづきポイント: Google Workspace のアカウント（会社のドメイン）で認証している場合、組織の IT ポリシーによってはサードパーティアプリの接続が制限されていることがあります。接続できない場合は、社内の IT 担当者に確認してください。

23.4.4 GitHub — リポジトリと連携

できること:

- リポジトリの README やコードを読む
- イシューの一覧・詳細を確認
- プルリクエストの内容を確認・サマリー
- イシューの作成・コメント

活用例（エンジニア・テックリード向け）:

このリポジトリの未解決のイシューを一覧にして、緊急度高そうなものをトップ 5 に絞って教えてください。

先週マージされたプルリクエストを確認して、主な変更内容をリリースノート形式でまとめてください。

必要な権限: リポジトリの読み取り（イシュー作成・コメントをする場合は書き込み権限も）

23.4.5 PowerPoint — プレゼン資料を AI で強化

できること:

- 競合分析スライドの自動生成（既存テンプレートのデザインを維持）
- スライド内の数値や通貨の一括変換（例: ドル → 円）
- デッキ全体の整合性チェック（CAGR 計算の矛盾検出など）

対応環境:

- PowerPoint on the web
- Windows 版（Microsoft 365 ビルド 16.0.13127.20296 以降）
- Mac 版（16.46 以降）
- ※ 永続ライセンス版（2016/2019）、iPad/Android は非対応

導入方法:

1. Microsoft AppSource から「Claude by Anthropic in PowerPoint」を検索してインストールします
2. Claude アカウントでサインインします
3. PowerPoint 内で Claude のパネルが表示され、スキル（コマンド）が使えるようになります

活用例:

`/competitive-analysis` — テンプレートを維持したまま競合分析デッキを生成

`/deck-refresh` — スライド内の数値やグラフを最新データに更新

`/ib-check-deck` — デッキ全体の数値の整合性を自動チェック

必要な権限: Microsoft 365 サブスクリプション。組織管理者による一括デプロイ（Microsoft 365 管理センター経由）にも対応しています。

つまづきポイント: 永続ライセンス版の PowerPoint（Office 2016、Office 2019 など）ではこのアドインを利用できません。Microsoft 365 サブスクリプション版が必要です。

23.4.6 Gmail — メールを自動処理

できること:

- 受信メールの読み取り・検索
- メールの送信
- 下書きの作成
- ラベル・フォルダの整理

活用例:

今日受信したメールを確認して、
返信が必要なものをリストアップして優先順位をつけてください。

以下の内容でメールを作成して、
draft として保存してください（送信はしないで）：
宛先: yamada@example.com
件名: 来週の打ち合わせについて
本文: ...

必要な権限: メールを読み取り（送信する場合は送信権限も）

つまづきポイント: Gmail プラグインはメールを自動送信する権限を持ちます。重要なメールを誤送信しないよう、Claude に「送信はせず draft として保存して」と指示してから内容を確認し、問題なければ手動で送信する習慣をつけることをおすすめします。

23.5 複数プラグインを組み合わせる

プラグインの真の力は、複数を組み合わせたときに発揮されます。

23.5.1 活用例 1: Slack → Notion に自動転記

Slack の #meeting チャンネルで今日投稿された議事録メモをすべて読んで、
Notion の「議事録」データベースに今日の日付で新しいページを作成し、
内容を整理して書き込んでください。

使用プラグイン: Slack + Notion

23.5.2 活用例 2: Google Drive → メールでレポート送付

Google ドライブの「月次レポート」フォルダにある最新の CSV ファイルを読んで、
先月比と傾向をまとめたレポートを作成し、
management@example.com に「月次レポート（〇月分）」という件名でメール送信してください。

使用プラグイン: Google Drive + Gmail

23.5.3 活用例 3: GitHub イシュー → Slack 通知

GitHub の「myproject」リポジトリで今週新しく作成されたイシューを確認して、
priority:high のラベルがついているものを Slack の #dev チャンネルに投稿してください。

使用プラグイン: GitHub + Slack

23.6 カスタムプラグインの作成（概要）

公式プラグインに含まれていないサービスや、社内システムと連携したい場合は、カスタムプラグインを作成できます。

23.6.1 カスタムプラグインとは

カスタムプラグインは、MCP サーバーという仕組みを使って自分でゼロから作る連携機能です。プログラミング（TypeScript または Python）の知識が必要になります。

23.6.2 どんなケースで必要になるか

- 社内独自のシステム（SFA、ERP など）と連携したい
- 公式プラグインにない API を使いたい
- 既存プラグインの動作をカスタマイズしたい

23.6.3 詳細は MCP ガイドへ

カスタムプラグインの作成手順・具体的なコード例・デバッグ方法などは、以下の専門ガイドで詳しく解説しています。

[MCP ガイド — Claude に外部ツールをつなげる](#)（ページ 345）

エンジニアやテックリードの方で、既存プラグインでは要件を満たせない場合はこちらを参照してください。

23.7 トラブルシューティング

23.7.1 プラグインが接続できない

症状: インストールしたのに「接続されていません」と表示される

対処法:

1. 「設定 > プラグイン」から対象プラグインを選択し、「再接続」をクリックする
2. 認証の有効期限が切れている場合は、もう一度ログイン（認証）し直す
3. それでも解決しない場合は、プラグインを一度削除して再インストールする

23.7.2 プラグインをインストールしたのに Claude が使ってくれない

症状: 「Slack を確認して」と頼んでも、Claude が「Slack にアクセスできません」と答える

対処法:

1. プラグインのトグルが「有効」になっているか確認する（設定 > プラグイン）
2. そのサービスへの権限が正しく付与されているか確認する
3. タスクの指示に「Slack プラグインを使って」と明記してみる

つまづきポイント: Claude は複数のプラグインがある場合、自動的に適切なプラグインを選ぶうとしますが、判断が曖昧なこともあります。「Slack プラグインを使って #general チャンネルを確認して」のように、使うプラグインを明示的に指定すると確実です。

23.7.3 認証エラーが繰り返し発生する

症状: 接続は成功したように見えるが、タスク実行中に「認証エラー」が出る

対処法:

1. 外部サービス (Slack、Notion など) 側でパスワードが変更されていないか確認する
2. そのサービスで「接続済みアプリ」の一覧を確認し、Claude の連携を一度削除してから再連携する
3. 組織アカウントを使っている場合は、IT 管理者にサードパーティアプリの接続が許可されているか確認する

23.7.4 アクセス権限が足りない

症状: 「そのファイルにアクセスする権限がありません」と表示される

対処法:

1. プラグインをインストールした際に、必要な権限 (例: 「書き込み」権限) を許可しなかった可能性があります
2. プラグインを削除して再インストールし、認証画面で必要な権限をすべて許可します

23.8 次のステップ

プラグインを活用して Cowork の可能性を広げましょう。

- **MCP ガイド — Claude に外部ツールをつなげる** (ページ 345) — カスタムプラグインの作成方法や MCP サーバーの詳細について学ぶ
- **Cowork 入門 — Claude にタスクを任せよう** (ページ 179) — プラグインを使ったスケジュールタスクや複合タスクの実践例
- **Claude Desktop アプリの使い方** (ページ 171) — Desktop アプリ全体の設定・管理方法
- **ビジネス活用事例** (ページ 209) — プラグイン連携を使った実践的なワークフロー例

第24章 Claude in Chrome — ブラウザ上で Claude に仕事を任せる

Claude in Chrome は、Google Chrome のサイドパネルに常駐する AI エージェント拡張機能です。ブラウザで開いたページを Claude が読み取り、クリックや入力を代わりに行うことで、日常のブラウザ作業をそのまま Claude に任せられます。

この記事では、インストールから実際の活用例まで順を追って解説します。

24.1 Claude in Chrome とは

Claude in Chrome をインストールすると、ブラウザの右端にサイドパネルが常駐します。Claude はあなたが見ているページをリアルタイムで把握し、以下のような操作を代行できます。

- **ページの読み取り・要約:** 開いているページの内容を即座に要約
- **クリック・入力の代行:** フォームへの入力、ボタンのクリック、メール送信
- **複数タブの管理:** 複数のタブを同時に扱うワークフローを実行
- **バックグラウンド実行:** タスクを依頼したまま別の作業ができる
- **スケジュール実行:** 毎日・毎週など定期的なタスクを自動化

ブラウザ上の操作を自動化する点では「Cowork」と似ていますが、Claude in Chrome はブラウザ内に特化したエージェントです。Cowork がデスクトップ全体を操作できるのに対し、Claude in Chrome は Web ブラウザの中だけで動きます。

24.1.1 対応ブラウザ



警告:

Claude in Chrome は **Google Chrome のみ**に対応しています。Microsoft Edge、Brave、Arc など他の Chromium ベースのブラウザや、モバイルデバイスでは利用できません。

24.1.2 対応プランとモデル

プラン	利用可否	使えるモデル
Free	利用不可	—
Pro	利用可（ベータ）	Haiku 4.5 のみ
Max	利用可	Opus 4.6 / Sonnet 4.6 / Haiku 4.5
Team	利用可	Opus 4.6 / Sonnet 4.6 / Haiku 4.5
Enterprise	利用可	Opus 4.6 / Sonnet 4.6 / Haiku 4.5

モデルの選び方の目安:

- **Haiku 4.5:** 素早い応答が欲しい場面・単純な操作
- **Sonnet 4.6:** 日常業務のほとんどに最適なバランス型
- **Opus 4.6:** 複雑な判断や推論が必要な場面

つまづきポイント: Free プランでは Claude in Chrome は利用できません。Pro 以上のプランへのアップグレードが必要です。プラン変更は claude.ai/settings から行えます。

24.2 インストールとセットアップ

24.2.1 インストール手順

1. **Chrome を開く** Google Chrome ブラウザを起動します
2. **Chrome Web Store にアクセスする** アドレスバーに `chrome.google.com/webstore` と入力してアクセスするか、Chrome Web Store で「Claude」を検索します
3. **「Add to Chrome」をクリックする** Claude の拡張機能を見つけたら「Chrome に追加」ボタンをクリックします
4. **権限を承認する** 「拡張機能を追加」ダイアログが表示されるので、確認してクリックします
5. **Claude アカウントでログインする** インストール後、Claude アカウント（有料プラン）でログインします
6. **拡張機能をピン留めする** Chrome のアドレスバー右端にある「パズルピース」アイコン（拡張機能メニュー）をクリックし、Claude の横にある「ピン」アイコンをクリックします。これで常にツールバーにアイコンが表示されます

24.2.2 権限モードの設定（重要）

Claude in Chrome には 2 種類の動作モードがあります。初めて使う場合は「Ask before acting」を強くおすすめします。

モード	動作	向いている場面
Ask before acting（推奨）	操作前に計画を表示し、承認を求める	慣れるまでの初期段階・重要な操作

次のページに続く

表 2 – 前のページからの続き

モード	動作	向いている場面
Act without asking	確認なしで自動実行（一部の高リスク操作は除く）	操作に十分慣れた後



警告:

「Act without asking」モードはより効率的ですが、Claude が予期しない操作を行う可能性があります。まずは「Ask before acting」モードで Claude の動作を確認してから、慣れてきたら切り替えることをおすすめします。

24.2.3 サイドパネルの開き方

インストール後は、以下の方法でサイドパネルを開けます。

- ツールバーの Claude アイコンをクリックする
- 右クリックメニューから「Claude を開く」を選択する

サイドパネルが表示されたら、準備完了です。

24.3 基本的な使い方

24.3.1 ページを読み取って質問する

最もシンプルな使い方は、開いているページについて質問することです。

1. 読みたいページを開きます（ニュース記事、商品ページ、PDF など）
2. サイドパネルの入力欄に質問を入力します
3. Claude が現在のページ内容を読み取って回答します

プロンプト例:

このページの内容を 3 行で要約して

このページに書かれている重要なポイントを箇条書きにして

このページの情報から、競合他社と比べた特徴を教えて

24.3.2 ページ内を操作させる（クリック・入力）

Claude にページ上の操作を任せることができます。「Ask before acting」モードの場合、実行前に Claude が計画を提示して確認を求めます。

1. やってほしいことを自然な言葉で伝えます
2. Claude が操作計画を提示します（Ask before acting モードの場合）

3. 内容を確認して「承認」または「キャンセル」を選択します
4. 承認すると Claude が実際に操作を実行します

プロンプト例（Gmail を開いた状態で）：

件名「打ち合わせのお礼」で田中さんへ御礼メールを送って。
内容は「先日はお時間をいただきありがとうございました。
今後ともよろしくお願いします。」

つまづきポイント: Claude が「ページを読み取れない」という場合は、拡張機能のアクセス許可を確認してください。サイドパネル上部の設定から、現在のサイトへのアクセスを「許可」に変更すると解決することが多いです。

24.3.3 ショートカット（保存済みプロンプト）

よく使う指示は「ショートカット」として登録できます。サイドパネルの入力欄で / を入力すると、保存済みのショートカット一覧が表示されます。

頻繁に使うプロンプト（例: 「要約して」「翻訳して」「アクションアイテムを抽出して」）をショートカットに登録しておくと、毎回入力する手間が省けます。

24.3.4 複数タブでの作業

Claude in Chrome は、複数のタブをまたいだ作業も処理できます。

プロンプト例（複数のタブを開いた状態で）：

今開いているタブ全部を見て、それぞれのページの概要を教えて

タブ 1（会議カレンダー）の予定を確認して、
タブ 2（Notion のメモ）に今週の予定一覧を追記して

24.4 活用例

24.4.1 メール作成（Gmail）

Gmail を開いた状態でサイドパネルに指示を出すと、メールの下書きや送信を代行できます。

基本の下書き作成:

件名「プロジェクト進捗報告」で鈴木部長宛に
今週の進捗状況メールを作成して。
プロジェクト名は「新製品開発」、進捗は 50%完了。
来週のマイルストーンも含めて。

返信の作成:

このメールに対して、ミーティングの承諾返信を書いて。
希望日時は来週火曜の 14 時。場所は弊社会議室 A。

受信トレイの整理:

未読メールの件名と送信者を一覧にして、
緊急度高そうなものを教えて

24.4.2 リサーチ・要約

複数のページを横断してリサーチする作業も得意です。

競合リサーチ:

この会社のサービスページを読んで、

- 主なサービス内容
- 料金体系
- 特徴・強み

をまとめた比較表を作って

ニュース収集:

今開いているニュースサイトの記事を 5 本読んで、
業界に関連するトピックをリストアップして

資料の要約:

この PDF（契約書）の重要な条項を
箇条書きで整理して。特に解約条件と
支払い条件に注目して。

24.4.3 データ抽出・入力

Web ページ上のデータを取り出して整理したり、フォームへの入力を自動化できます。

テーブルデータの抽出:

このページの料金表を読み取って、
CSV 形式でまとめて

求人情報の収集:

この求人一覧から、勤務地が東京・経験不問の案件を
全部抜き出して、会社名・職種・給与をリストにして

フォーム入力の自動化:

このお問い合わせフォームに以下の情報を入力して：

会社名：〇〇株式会社

担当者名：山田太郎

メール：yamada@example.com

お問い合わせ内容：[後でまとめてもらう]

つまづきポイント：フォームへの自動入力を行う場合は、実行前に内容を必ず確認してください。「Ask before acting」モードであれば、Claude が入力内容を提示してから実行するので安心です。

24.4.4 スケジュール管理

Google カレンダーなどのカレンダーサービスと連携して、スケジュールの確認や登録ができます。

Google カレンダーを開いて、今週の予定を一覧にして

「〇〇社との打ち合わせ」というイベントを
来週水曜の 15 時～16 時で追加して。
場所は「オンライン（Zoom）」で。

24.4.5 マルチサイトワークフロー

複数のサイトをまたいだ複合タスクも処理できます。

例：競合調査から報告書作成まで

A 社のサービスページ、B 社のサービスページを順番に見て、
料金・機能・特徴を比較した表を Notion の「競合調査」ページに追記して

例：リード管理の自動化

今開いているお問い合わせフォームの送信履歴を読んで、
CRM (Salesforce) の「新規リード」として登録して

24.4.6 スケジュール実行（定期タスク）

繰り返し行う調査タスクなどは、スケジュール実行に設定できます。

- 毎日・毎週・毎月などの頻度を選択
- 対象サイトとやってほしいことを指定
- 実行結果はサイドパネルやメール等で通知

設定例：

毎週月曜の朝 9 時に競合他社の新着ブログを確認して、
重要そうな記事のタイトルと概要をまとめて通知して

24.5 Claude Code との連携（開発者向け）



注釈:

このセクションはウェブ開発者向けの内容です。プログラミングに詳しくない方は、このセクションを飛ばして次のセクションへ進んでください。

Claude Code（ターミナルで使う開発者向けの Claude）と Claude in Chrome を連携させることで、コードを書いてすぐにブラウザで動作確認するワークフローを構築できます。

24.5.1 起動方法

ターミナルで以下のコマンドを実行します。

```
claude --chrome
```

または Claude Code の会話中に以下のスラッシュコマンドを使います。

```
/chrome
```

24.5.2 開発ワークフローの例

1. ターミナルで `claude --chrome` を起動
2. コードの修正を依頼する（例: 「このボタンのスタイルを変えて」）
3. Claude がコードを修正し、ブラウザで自動的にプレビューを確認
4. コンソールエラーや DOM の状態を読み取って次の修正に反映
5. 変更の様子を GIF として録画することも可能

24.5.3 Claude Code 連携時の制限

- Amazon Bedrock 経由では利用不可
- Google Vertex AI 経由では利用不可
- Microsoft Azure AI Foundry 経由では利用不可

24.6 Cowork との使い分け

Claude in Chrome と Cowork はどちらも「Claude にタスクを自動実行させる」機能ですが、役割が異なります。

項目	Claude in Chrome	Cowork
動作範囲	ブラウザ内のみ	デスクトップ全体
ファイル操作	ダウンロード・保存のみ	ローカルファイルの読み書き全般
対象	Web サイト・Web アプリ	PC 内のファイル・フォルダ・外部ツール
必要なもの	Chrome ブラウザ	Claude Desktop アプリ
向いている用途	Web 上の操作・情報収集	ファイル整理・ローカルデータ処理

24.6.1 どちらを使うべきか

Claude in Chrome が向いている場合:

- Gmail、Google カレンダー、Slack などの Web アプリ上での操作
- ウェブサイトの情報収集・要約・比較
- オンラインフォームへの入力

Cowork が向いている場合:

- PC 内の大量のファイルをまとめて処理する
- ローカルに保存された Excel や Word の操作
- Web とローカルファイルを組み合わせた処理（Cowork が Chrome を「手足」として使える）



注釈:

Cowork を使いながら Claude in Chrome を呼び出すこともできます。たとえば「ローカルの Excel データを読んで → Web の CRM に入力する」ような複合タスクでは、Cowork と Chrome を組み合わせると強力です。

24.7 安全に使うために

Claude in Chrome はブラウザ上で多くの操作を代行できるため、セキュリティに注意して使うことが重要です。

24.7.1 サイト別アクセス制御

Claude in Chrome は、サイトごとにアクセスを制御できます。

- **1 回限りの許可:** 今回の操作だけ許可する
- **常時許可:** このサイトはいつでも操作できる
- **拒否:** このサイトには操作させない

信頼できるサイトには「常時許可」、初めてのサイトは「1 回限りの許可」から始めるのがおすすめです。

24.7.2 デフォルトでブロックされるサイト

以下のカテゴリのサイトは、デフォルトでアクセスがブロックされています。

- 銀行・金融機関のサイト
- 投資・株式取引のプラットフォーム
- 暗号通貨取引所
- アダルトコンテンツサイト
- 海賊版コンテンツサイト

24.7.3 推奨しない用途

以下の用途には Claude in Chrome を使わないことを推奨します。

用途	理由
銀行・投資口座の操作	誤送金・誤操作のリスク
法的文書への署名・送信	内容の誤解釈による損失リスク
医療情報の入力・送信	誤情報による健康被害リスク
他人の個人情報を含む操作	プライバシー保護の観点

つまづきポイント: Claude に操作を任せる前に、「Ask before acting」モードで Claude の計画を確認する習慣をつけましょう。「なんか違う」と思ったら、実行前にキャンセルできます。

24.7.4 プロンプトインジェクション対策

悪意のあるウェブページが Claude に意図しない操作をさせようとする攻撃（プロンプトインジェクション）に対して、Anthropic は以下の対策を実施しています。

- モデルの学習段階での対策（Claude Opus 4.5 では攻撃成功率が約 1% まで低下）
- 悪意のある指示を検出するコンテンツフィルター
- JavaScript が実行可能なサイトのドメイン単位での承認制御

とはいえ完全にリスクをゼロにすることはできません。見知らぬサイトで操作を許可する際は注意してください。

24.7.5 Team / Enterprise 管理者向け

組織での利用時は、管理者が以下の設定を行えます。

- **許可リスト:** アクセスを許可するサイトの指定
- **ブロックリスト:** アクセスを禁止するサイトの指定
- **組織全体でのオン/オフ切り替え**

プラン	デフォルト設定
Team	デフォルト有効
Enterprise	デフォルト無効（管理者が有効化）

24.8 プランごとの違い

プラン	月額料金	Claude in Chrome	使えるモデル	管理者設定
Free	無料	利用不可	—	—
Pro	\$20	ベータ利用可	Haiku 4.5 のみ	—
Max 5x	\$100	利用可	Opus / Sonnet / Haiku	—
Max 20x	\$200	利用可	Opus / Sonnet / Haiku	—
Team	\$30/人	利用可	Opus / Sonnet / Haiku	有（許可/ブロックリスト）
Enterprise	要問い合わせ	利用可（デフォルト無効）	Opus / Sonnet / Haiku	有（組織全体の制御）



注釈:

Pro プランでは現在ベータとして Haiku 4.5 のみが利用可能です。より高性能なモデル（Sonnet・Opus）を使いたい場合は Max または Team プランへのアップグレードをご検討ください。

24.9 次のステップ

Claude in Chrome の基本を理解できたら、他の製品や機能との組み合わせも試してみましょう。

- [Claude Desktop アプリの使い方](#)（ページ 171）— デスクトップアプリと Chrome 拡張の使い分けを知る
- [Cowork 入門 — Claude にタスクを任せよう](#)（ページ 179）— ローカルファイルを含む自動化タスクを学ぶ
- [Claude Web \(claude.ai\) の使い方](#)（ページ 89）— ブラウザ版 Claude の基本機能を確認する
- [料金プランの比較](#)（ページ 15）— 自分に合ったプランを選ぶ

第Ⅲ部

活用ガイド

第25章 ビジネス活用ユースケース

「AIを使ってみたいけど、自分の仕事にどう活かせばいいかわからない」——そんな疑問を持つビジネスパーソンに向けて、Claudeを使った具体的な業務改善の例を紹介します。

プログラミングの知識は一切不要です。普段のチャットと同じように文章を入力するだけで、メール作成・データ分析・リサーチ・資料作成など、幅広い業務に活用できます。

25.1 ビジネスで Claude を使うメリット

Claude を業務に取り入れると、主に3つの面で効果が出ます。

25.1.1 コスト削減

外部の翻訳会社や編集者への依頼、コンサルタントへの調査依頼など、これまで費用がかかっていた作業を社内でこなせるようになります。月額\$20（約3,000円）のProプランで、翻訳・リサーチ・文書作成の外注コストを大幅に削減した事例が多く報告されています。

25.1.2 時間短縮

報告書の下書き、会議の議事録整理、メールの文案作成など、「考えれば書けるが時間がかかる」作業が数分で完了します。1日2〜3時間かかっていた文書作業が30分以内に収まるケースも珍しくありません。

25.1.3 品質向上

文章の誤字脱字チェック、論理の一貫性確認、競合との比較分析など、一人では見落としがちな視点を補ってくれます。

注意: Claude はあくまで「アシスタント」です。生成された内容は必ず人間が確認・修正してから使用してください。特にデータや数値を扱う場合は、情報源を確認する習慣をつけましょう。



注釈:

無料プランと有料プランの違い

基本的な会話・文書作成・翻訳は **Free プラン** でも利用できます。ただし、Deep Research、大量のファイルアップロード、優先アクセスなどの機能は **Pro プラン** 以上が必要です。詳しくは [料金プラン比較](#) (ページ 15) を参照してください。

25.2 メール・文書作成

25.2.1 よくある課題

「取引先への謝罪メールをどう書けばいいかわからない」「英語でのビジネスメールに自信がない」「月次報告書の文章を毎回一から書くのが大変」——こうした悩みは Claude で解決できます。

25.2.2 プロンプト例：謝罪メールの下書き

以下の状況に合わせた取引先への謝罪メールを書いてください。

状況：製品の納期が 1 週間遅延することが判明した
相手：長年取引のある大手小売チェーンの仕入れ担当者
トーン：誠実で丁寧、関係を維持することを重視
要件：
- 件名も提案してください
- 遅延の原因（部材調達の遅れ）を簡潔に説明
- 代替案（分割納品の提案）を含める
- 200～300 文字程度

25.2.3 結果のポイント

Claude はこの指示から、件名・本文・結び言葉まで一貫したトーンのメールを生成します。生成後は「もう少し謝罪の気持ちを強めて」「冒頭の挨拶を変えて」など、会話を続けながら調整できます。

25.2.4 Before / After の例

Before（自分で書いた下書き）

件名：納期について佐藤様、お世話になっております。今回の件ですが納期が遅れることになりました。申し訳ありません。できるだけ早く対応します。

After（Claude を使って改善）

件名：【ご連絡】〇〇製品の納期変更のお願い佐藤様、平素より大変お世話になっております。この度は弊社製品の納期に関して、誠に恐れ入りますがご連絡申し上げます。部材調達の遅れにより、予定納期より 1 週間遅延が生じる見込みとなりました。代替案として、先行可能な数量の分割納品も検討できます。何卒ご検討いただけますと幸いです。

25.3 データ分析・レポート

25.3.1 よくある課題

「Excel のデータがあるが、傾向を読み取って文章にするのが苦手」「グラフを作ったが、何を示すレポートを書けばいいかわからない」——Claude はデータを読み込んで分析し、レポート本文まで生成できます。

25.3.2 プロンプト例：売上データの分析

CSV ファイルや Excel ファイルを claude.ai にアップロードし、以下のように依頼します。

添付の売上データを分析して、以下をまとめてください。

1. 今月の売上トップ 3 商品と前月比
2. 売上が落ちている商品と、考えられる原因の仮説
3. 来月に向けた改善提案（2〜3 点）
4. 結果を棒グラフで可視化してください（Artifacts として表示）

対象読者：経営層（専門用語は避け、数字で語る）

25.3.3 Artifacts との連携

「棒グラフで可視化してください」と依頼すると、Claude が右側のパネル（Artifacts）にインタラクティブなグラフを生成します。グラフの色・軸ラベル・並び順なども会話で調整できます。

注意（機密情報について）：売上データ・顧客リスト・個人情報などを Claude に貼り付ける場合は、社内の情報管理ポリシーを必ず確認してください。特に個人が特定できる情報（氏名・メールアドレス等）は事前に匿名化することを強くおすすめします。Claude に入力した内容はデフォルトでは学習に使用されませんが、社内規定によっては外部サービスへの入力自体が制限される場合があります。

25.3.4 プロンプト例：月次レポートの自動生成

以下のデータをもとに、経営会議向けの月次レポートを作成してください。

【今月のデータ】

- 売上高：2,340 万円（前月比+8%、前年同月比+15%）
- 新規顧客数：47 社
- 解約数：3 社（チャーンレート：0.8%）
- 主なトピック：新製品ライン C の受注が好調。一方、既存製品 A の競合からの価格圧力が増している

形式：

- A4 1 ページに収まる量
- 冒頭にエグゼクティブサマリー（3 行以内）
- 数値は表形式で
- リスクと対策を必ず含める

25.4 リサーチ・情報収集

25.4.1 よくある課題

「業界の最新トレンドを調べてレポートにまとめたいが、どこから手をつければいいのかわからない」「競合他社の動向を定期的にまとめる時間がない」

25.4.2 Deep Research の活用

Claude Web の Pro 以上のプランでは、「Deep Research」機能を使って数十分かけて複数の情報源を横断調査できます。

適したリサーチテーマの例:

- ・「2026 年の国内 SaaS 市場規模と主要プレイヤーの動向をまとめて」
- ・「競合 A 社・B 社・C 社の料金体系と機能を比較した表を作って」
- ・「電子帳簿保存法の 2024 年改正で変わった主な要件を整理して」

25.4.3 通常会話でのリサーチ例

Deep Research を使わなくても、Claude の知識を活用したリサーチは十分効果的です。

SWOT 分析のフレームワークを使って、以下のビジネス状況を整理してください。

【状況】

- 業種: 中規模の食品メーカー（従業員 200 名）
- 課題: EC チャネルへの参入を検討中
- 競合: 大手食品メーカーが EC に積極投資している
- 強み: 特定地域でのブランド認知度が高い
- 懸念: 物流コストと返品対応

SWOT の各項目を箇条書きで整理し、最後に参入にあたって最優先で対処すべき課題を 1 つ提案してください。

つまづきポイント: Claude の知識には学習データのカットオフ（知識の締め切り日）があります。最新の株価・為替・法改正情報など、リアルタイム性が重要なデータは公式サイトや官公庁のウェブサイトです。必ず確認しましょう。Deep Research 機能はウェブ検索ができるため、最新情報の調査に向いています。

25.5 プレゼン資料の作成

25.5.1 よくある課題

「プレゼンの構成をどう組み立てればいいかわからない」「スライドに入れる文章が長くなりすぎる」「経営層向けと現場向けで資料を作り分けるのが大変」

25.5.2 プロンプト例：プレゼン構成案の作成

以下の内容で経営会議向けプレゼンの構成案を作ってください。

テーマ: 新規事業として EC サイト立ち上げを提案する
対象: 社長・取締役 3 名（数字に厳しく、リスクを重視する）
持ち時間: 20 分（質疑応答含む）

(次のページに続く)

(前のページからの続き)

目的: 予算 3,000 万円の承認を得る

出力:

- スライド枚数の提案 (全体構成)
- 各スライドのタイトルとキーメッセージ (1 行)
- 特に重点を置くべきスライドのアドバイス

25.5.3 プロンプト例：スライドの本文テキスト生成

構成案ができたなら、各スライドの文章を生成してもらいましょう。

以下の条件でスライド「投資対効果 (ROI) 試算」の本文テキストを書いてください。

前提:

- 初期投資: 3,000 万円
- 3 年後の売上目標: 年間 1.5 億円
- 粗利率: 30%

スライドの構成:

- タイトルと 1 行サマリー
- 前提条件 (箇条書き 3 点)
- ROI の計算式と結果
- リスクシナリオ (保守的・標準・楽観的の 3 ケース)
- 結論の一文

1 スライドに収まるよう、文章は簡潔に (各項目 2~3 行以内)

つまづきポイント: Claude が生成した数値や試算はあくまで「たたき台」です。実際に使用する前に、財務担当者やマネージャーと数字を確認しましょう。特に「Claude が計算しました」と説明するのではなく、あなた自身の判断として内容を把握した上で発表することが重要です。

25.6 翻訳・多言語対応

25.6.1 よくある課題

「英語のメールを翻訳ツールで訳したが、ニュアンスがおかしい気がする」「日本語の提案書を海外本社向けに英訳したいが、専門用語の扱いに困っている」

25.6.2 プロンプト例：ビジネス文書の翻訳

以下の日本語の提案書を英語に翻訳してください。

【翻訳の条件】

- 対象: 米国本社を経営層 (日本ビジネスの文化に不慣れな可能性がある)

(次のページに続く)

(前のページからの続き)

- トーン: フォーマルなビジネス英語
- 専門用語の扱い: 初出時に括弧で英語の一般的な表現を付記する
- 曖昧な日本語表現は、意図を推測して明確な英語に変換する
- 翻訳後に「翻訳で工夫した点」を 3 つ箇条書きで教えてください

【翻訳対象テキスト】

(ここに日本語テキストを貼り付け)

25.6.3 用語集を活用したブレない翻訳

同じ製品・サービスに関する文書を複数回翻訳する場合は、用語統一が重要です。

以下の用語対応表を必ず守って翻訳してください。

【用語対応表】

- 「取引先」= "business partner" ("client" や "vendor" は使わない)
- 「稟議」= "internal approval process"
- 「案件」= "opportunity" (セールス文脈) または "project" (作業文脈)

【翻訳対象テキスト】

(ここにテキストを貼り付け)

25.7 経理・財務

25.7.1 よくある課題

「Excel の数式が複雑でわからない」「仕訳の勘定科目が正しいか自信がない」「予算シミュレーションを作りたいが、表の設計から困っている」

25.7.2 プロンプト例: Excel 数式の生成

Excel で以下の計算をする数式を教えてください。

目的: 月次売上データから、前年同月比の成長率を計算したい

シート構成:

- A 列: 月 (1~12 月)
- B 列: 今年の売上
- C 列: 昨年の売上
- D 列: ここに前年同月比 (%) を計算したい

条件:

- 昨年の売上が 0 の場合はエラーが出ないようにする

(次のページに続く)

(前のページからの続き)

- 成長率はパーセント表示（例：+12.5%）
- D2 セルに入れる数式と、他のセルへのコピー方法も教えてください

25.7.3 プロンプト例：仕訳の確認

以下の取引の仕訳を確認してください。

取引：ソフトウェアの年間ライセンス料として 120 万円を一括で支払った（4 月 1 日）

支払方法：銀行振込

私の仕訳案：

（借方）ソフトウェア使用料 1,200,000 / （貸方）普通預金 1,200,000

質問：

1. この仕訳は適切ですか？
2. 前払費用として処理すべき場合、どう変わりますか？
3. 消費税の処理についても教えてください

注意（税務・法務について）： Claude が提示する仕訳や税務の情報は参考情報です。確定申告・税務申告・重要な契約書類への使用は、必ず税理士・公認会計士・弁護士などの専門家に確認してください。法律や税制は改正されることがあり、Claude の回答が最新でない可能性があります。

25.7.4 プロンプト例：予算シミュレーション表の設計

新規事業の 3 年間予算シミュレーション表を Excel で作りたいです。

事業概要：

- SaaS 型の業務管理ツールを月額制で提供
- 初年度は無料トライアル中心でユーザーを獲得し、2 年目から課金転換

含めてほしい項目：

- 収益：月次ユーザー数・課金転換率・月額単価
- 費用：人件費・サーバー代・マーケティング費・その他
- KPI：MRR（月次経常収益）・チャーンレート・CAC（顧客獲得コスト）・LTV

出力：

- 表の構成（行・列の設計）
- 各セルで使う Excel 数式の例
- 見やすくするための書式設定のアドバイス

25.8 営業・顧客対応

25.8.1 よくある課題

「提案書を毎回ゼロから作るのが大変」「顧客からよく来る質問の FAQ をまとめたい」「商談後のフォローメールが形式的になっている」

25.8.2 プロンプト例：提案書のカスタマイズ

以下の標準提案書テンプレートを、顧客の状況に合わせてカスタマイズしてください。

【標準テンプレート（概要）】

（ここに標準提案書の本文を貼り付け）

【顧客情報】

- 会社名：△△株式会社（製造業、従業員 500 名）
- 担当者：情報システム部 鈴木部長（コスト削減に強い関心）
- 課題：現在の基幹システムが古く、リプレースを検討中
- 予算感：年間 1,500 万円以内
- 競合状況：B 社・C 社と比較中

カスタマイズ方針：

- コスト削減効果の数字を前面に出す
- 他 2 社との差別化ポイントを強調
- 鈴木部長が意思決定しやすいよう、ROI 試算を追加

25.8.3 プロンプト例：FAQ の作成

以下の製品仕様書をもとに、よくある質問（FAQ）を作成してください。

【製品仕様書】

（ここに仕様書を貼り付け）

FAQ の要件：

- 想定読者：製品を初めて検討する企業の購買担当者
- 質問数：10～15 問
- 各回答は 200 文字以内
- カテゴリ分け：機能・料金・サポート・導入・セキュリティ
- 「よくある誤解」を払拭する FAQ も含める

25.8.4 プロンプト例：商談後フォローメール

商談後のフォローメールを書いてください。

(次のページに続く)

(前のページからの続き)

商談の概要:

- 日時: 今日の 14 時～15 時 (Web 会議)
- 相手: 山田商事 営業部 佐々木様
- 商談内容: 弊社の在庫管理システムのデモを実施。先方から「使いやすい」と好感触
- 宿題事項: 弊社側から見積書 (来週月曜までに)、先方から現行システムの課題リストを共有予定

メールの条件:

- 商談への感謝と要点の確認
- 見積書の送付予定を明記
- 次のステップを明確に
- 温かみがありつつビジネスライクなトーン

つまづきポイント: プロンプトに顧客の実名・会社名・具体的な金額などの機密情報を含める場合は、社内の情報セキュリティポリシーを事前に確認してください。必要に応じて「A 社」「担当者 B」など仮の名前に置き換えてから入力し、生成後に実際の情報に差し替える方法も有効です。

25.9 企業での導入事例

Claude は個人の業務効率化だけでなく、企業全体での導入も進んでいます。

25.9.1 企業での採用例

- **Intercom** (カスタマーサポート) : Claude を活用した AI エージェント「Fin」で、顧客からの問い合わせに対して**最大 86% の自動解決率**を達成。応答時間は 30 分から数秒に短縮し、45 以上の言語に対応しています
- **Asana** (プロジェクト管理) : Claude を自社のワーク管理プラットフォームに統合。開発サイクルが数年規模から**数週間に短縮**され、データに基づく意思決定が 10 倍速くなったと報告しています
- **Binti** (社会福祉) : ソーシャルワーカーがレポート作成に費やす時間を**半減**。書類仕事の負担を減らし、本来の支援業務に集中できるようになりました

25.9.2 まずは小さく始める

大企業の事例は規模が大きいです。中小企業や個人での活用も同じ考え方で始められます。

1. **1つの業務から始める:** まずは「議事録の要約」「メールの下書き」など、1つの定型作業を Claude に任せてみる
2. **効果を実感したら範囲を広げる:** 「Excel の分析」「提案書の作成」など、対象業務を徐々に広げる
3. **チームで共有する:** うまくいったプロンプトやワークフローをチームメンバーに共有する

25.10 機密情報の取り扱いについて

Claude をビジネスで活用する際、最も気をつけたいのが情報セキュリティです。

25.10.1 入力してよい情報・避けるべき情報

入力してよい情報の例	注意が必要な情報
架空のデータ・サンプルデータ	顧客の実名・個人情報
公開されている業界情報	未発表の製品・サービス情報
自社の公開済み資料	M&A・組織再編などの機密事項
一般的なビジネス文書のテンプレート	内部の価格・原価・利益率

25.10.2 Claude のデータ利用ポリシー

Anthropic の方針では、claude.ai での会話は原則としてモデルの学習には使用されません（2024 年時点）。ただし、企業での利用の際は以下を確認することを推奨します：

- 社内のクラウドサービス利用ポリシーを確認する
- 機密性の高い情報は Anthropic の API を使った企業向けプラン（Claude for Business）の利用を検討する
- データの取り扱いについて疑問がある場合は、法務・情報システム部門に相談する

25.11 次のステップ

このページで紹介したユースケースをさらに深めるには、以下の記事も参考にしてください。

- [Claude Web \(claude.ai\) の使い方](#) (ページ 89) --- 基本操作から Artifacts・Deep Research まで
- [プロンプトの基本テクニック](#) (ページ 239) --- Claude に伝わる指示の書き方を学ぶ
- [マーケティング活用ユースケース](#) (ページ 229) --- コンテンツ制作・SNS 運用・市場分析に特化した活用例
- [エンジニアリング活用ユースケース](#) (ページ 219) --- コード生成・レビュー・ドキュメント作成

第26章 エンジニアリング活用ユースケース

Claude はソフトウェア開発のあらゆるフェーズで活用できます。コードを書くだけでなく、設計の壁打ち、バグの調査、テストの生成、ドキュメントの整備まで、開発ワークフロー全体を加速してくれるパートナーです。

この記事では、エンジニアが日常業務で Claude をどのように使えるかを、具体的なプロンプト例とコマンド例とともに紹介します。

26.1 エンジニアリングで Claude を使うメリット

26.1.1 生産性の向上

繰り返しの多い作業（ボイラープレートコードの生成、テストの作成、ドキュメントの整備）を Claude に任せることで、本質的な設計や問題解決に集中できます。

- ユニットテストを書く時間が大幅に短縮される
- 初めて触る API やライブラリのキャッチアップが速くなる
- ドキュメント作成の心理的ハードルが下がる

26.1.2 学習の加速

Claude は「なぜそうなるのか」を説明しながら教えてくれます。Stack Overflow で断片的な答えを探すより、自分のコードの文脈に沿った解説が得られます。

- 新しい言語やフレームワークに慣れるのが速くなる
- 既存コードベースを引き継いだときの把握が速くなる
- アーキテクチャの選択肢をすばやく比較できる

26.1.3 品質の向上

Claude をコードレビューアとして活用することで、見落としがちなバグ、セキュリティ上の問題点、改善できる設計を指摘してもらえます。

どの Claude を使えばいいか



注釈:

ユースケース	推奨ツール
コーディング・ファイル操作	Claude Code（ターミナル）
設計の壁打ち・議論	Claude Web / Desktop
CI/CD への組み込み、自動化	API
IDE でコードを書きながら	VS Code / JetBrains 統合

26.2 コーディング支援

26.2.1 コード生成とリファクタリング

Claude Code を使うと、自然言語でコードを書かせたり、既存コードを改善させたりできます。

プロジェクトのディレクトリで Claude Code を起動して指示する例：

```
claude
```

```
> src/utils/date.ts に日付フォーマット関数を追加して。
ISO 8601 形式の文字列を受け取り、"2024 年 1 月 15 日" 形式で返す関数。
TypeScript の型定義も含めて。
```

```
> src/components/Button.tsx をリファクタリングして。
現在 props が増えすぎているので、variant パターンを使って整理したい。
```

リファクタリングを依頼するときは、変更の方針や制約と一緒に伝えると精度が上がります。

```
> src/api/users.ts の fetchUsers 関数をリファクタリングして。
以下の制約で：
- 外部インターフェースは変えない
- エラーハンドリングを強化する
- 型定義を厳密にする（any を使わない）
```

26.2.2 IDE 統合での使い方

VS Code や JetBrains IDEs では、エディタ内から Claude に直接コード補完や説明を依頼できます。コードを書きながらその場で質問できるため、コンテキストの切り替えが不要です。

詳しくは [IDE 統合ガイド](#)（ページ 159）を参照してください。

つまづきポイント: Claude に「このコードを直して」とだけ伝えると、期待と違う変更をされることがあります。「どこを、どのように直してほしいか」を明確に伝えましょう。たとえば「パフォーマンスを改善して。ただし処理の結果は変えないで」のように制約を添えると意図通りの結果になりやすいです。

26.3 デバッグ

26.3.1 エラーメッセージの解析

エラーが出たとき、メッセージとスタックトレースをそのまま Claude に渡すと、原因と解決策を教えてください。

```
# エラーログをそのまま渡す
cat error.log | claude -p "このエラーの原因を特定して、修正方法を教えて"

# ビルドエラーを解析する
pnpm build 2>&1 | claude -p "このビルドエラーを解析して、修正手順を教えて"
```

Claude Code の対話モードで直接貼り付けることもできます。

> 以下のエラーが出て困っています。原因と対処法を教えてください:

```
TypeError: Cannot read properties of undefined (reading 'map')
    at UserList (/src/components/UserList.tsx:24:23)
    at renderWithHooks (/node_modules/react-dom/...)
```

Claude はエラーの種類を特定し、「users が undefined になっているケースがある」「初期値を空配列にすべき」などの具体的な修正案を提示してくれます。

26.3.2 ログ分析と根本原因の特定

長いログファイルや複雑な障害調査も Claude が助けてくれます。

```
# アプリケーションログの異常を検出する
tail -n 1000 /var/log/app.log | claude -p "このログから異常なパターンを見つけて、根本原因の仮説を立てて"

# 特定の期間のエラーを集計する
grep "ERROR" app.log | claude -p "エラーの種類別に集計して、最も頻度が高いものから順に並べて"
```

つまづきポイント: ログに個人情報や機密情報が含まれている場合は、Claude Web (claude.ai) や API に送らないよう注意してください。ローカルで動く Claude Code であれば、コードはローカルのファイルシステムに直接アクセスするため、外部に送るデータをコントロールしやすいです。ただし Claude Code も会話内容は Anthropic のサーバーに送られます。詳しくは **セキュリティと機密情報の取り扱い** (ページ 226) を参照してください。

26.4 コードレビュー

26.4.1 PR レビューの自動化

コードをコミットする前に Claude にレビューさせることで、セルフレビューの質が上がります。

```
# 現在の変更をレビューしてもらう
git diff | claude -p "このコードの変更をレビューして。バグ、セキュリティ上の問題、改善できる点を指摘して"

# 特定のファイルをレビューする
cat src/api/auth.ts | claude -p "このファイルをセキュリティの観点でレビューして"

# PR のタイトルとコミットメッセージを生成する
git diff main | claude -p "この変更の内容をもとに、PR のタイトルと説明文を日本語で書いて"
```

Claude Code の対話モードでは、ファイルを直接参照しながらレビューを依頼できます。

```
> @src/api/users.ts このファイルをレビューして。
特に以下の観点で確認してほしい：
1. 入力値のバリデーションが適切かどうか
2. エラーハンドリングが十分かどうか
3. SQL インジェクションの危険がないかどうか
```

26.4.2 セキュリティチェック

セキュリティの脆弱性チェックも Claude が助けてくれます。

```
> 認証周りのコードをレビューして。OWASP Top 10 の観点で問題がないか確認したい。
対象ファイル: src/middleware/auth.ts, src/api/login.ts
```

```
> @src/utils/sanitize.ts XSS 対策が適切かどうかを確認して。
攻撃パターンの例も一緒に教えてほしい。
```

つまづきポイント: Claude はコードの問題点を指摘してくれますが、セキュリティ診断ツールの代替にはなりません。本番環境に影響するセキュリティ上の変更は、必ず専門家のレビューや自動スキャンツール（SAST/DAST）と組み合わせて確認してください。

26.5 設計・アーキテクチャ

26.5.1 システム設計の壁打ち

実装を始める前に Claude と設計を議論することで、後から大きく変更することを防げます。Claude Web (claude.ai) が会話形式で設計を深めやすく、向いています。

プロンプト例:

```
新しい通知システムを設計したい。
要件：
- ユーザーにメール・プッシュ通知・SMS を送れる
- 通知の種類は今後増やせるようにしたい
- 大量配信（1 万件以上）にも対応したい
```

(次のページに続く)

(前のページからの続き)

技術スタック: Node.js, PostgreSQL, AWS

どんなアーキテクチャが適切か、選択肢をいくつか提示してほしい。
それぞれのトレードオフも教えてほしい。

26.5.2 技術選定の比較

ライブラリやフレームワークを選ぶときも Claude が助けてくれます。

React の状態管理ライブラリを選びたい。

候補: Redux Toolkit, Zustand, Jotai

プロジェクトの条件:

- チームの人数: 5 人
- アプリの規模: 中規模 SPA
- テストのしやすさを重視したい
- 学習コストは低い方がいい

それぞれの特徴と、この条件に合う選択肢を教えてください。

26.5.3 設計ドキュメントの作成

既存の実装から設計ドキュメントを自動生成することもできます。

システム構成の説明ドキュメントを生成

claude -p "src/ ディレクトリを読んで、このシステムのアーキテクチャ概要ドキュメントを、
↪Markdown 形式で生成して"

> このプロジェクトの src/api/ ディレクトリを読んで、
API の設計について ADR (Architecture Decision Record) を書いて。
特にルーティングの方針と認証の仕組みについて記録してほしい。

26.6 テスト

26.6.1 ユニットテストの自動生成

テストを書くのは時間がかかりますが、Claude に任せることで大幅に効率化できます。

特定のファイルのテストを生成する

cat src/utils/format.ts | claude -p "このファイルの関数に対するユニットテストを Vitest で
書いて。エッジケースも網羅してほしい"

Claude Code では対話しながらテストを育てていきます。

```
> @src/services/userService.ts このサービスのユニットテストを書いて。  
使用フレームワーク: Vitest  
モックライブラリ: vi.fn()  
外部 API の呼び出しはすべてモックしてほしい
```

```
> 生成してくれたテストを見ると、ユーザーが存在しない場合のエラーケースが足りていない。  
追加してほしい。
```

26.6.2 テストケースの網羅性チェック

既存のテストが十分かどうか確認できます。

```
> @src/utils/__tests__/date.test.ts 現在のテストケースを確認して、  
カバーできていないエッジケースを教えて。  
特に:  
- タイムゾーンの扱い  
- 無効な日付の入力  
- 境界値 (年またぎ、うるう年など)
```

26.6.3 E2E テストシナリオの作成

Playwright や Cypress を使った E2E テストのシナリオも Claude が書けます。

```
> ユーザーログインフローの E2E テストを Playwright で書いて。  
シナリオ:  
1. 正常ログイン (正しいメールとパスワード)  
2. パスワード誤り (3 回失敗でロック)  
3. パスワードリセットフロー  
  
ベース URL: http://localhost:3000
```

つまづきポイント: Claude が生成したテストは、必ず実際に動かして確認してください。生成されたテストが実装の仕様を正確に反映していない場合があります。特に「このテストは通るべきか、落ちるべきか」を確認しながら使うと品質が上がります。

26.7 ドキュメント

26.7.1 README の生成

プロジェクトの概要を Claude Code に読ませて README を自動生成できます。

```
claude -p "このプロジェクトを全体的に読んで、わかりやすい README.md を生成して。  
インストール手順、基本的な使い方、設定方法を含めてほしい"
```

26.7.2 API ドキュメントの生成

コードから API の仕様書を生成することもできます。

```
> src/api/ ディレクトリのエンドポイントを読んで、
  OpenAPI 3.0 形式の仕様書を生成して。
  各エンドポイントのパラメータ、レスポンス形式、エラーコードを網羅してほしい。
```

26.7.3 コードコメントの追加

コードにコメントを追加する作業も自動化できます。

```
> @src/utils/crypto.ts この暗号化ユーティリティに JSDoc コメントを追加して。
  各関数の用途、引数の説明、戻り値の型、使用例をコメントで記載してほしい。
```

26.7.4 ADR (Architecture Decision Record) の生成

設計の決定を記録する ADR も Claude が書けます。

```
> 先日の議論で、状態管理に Redux Toolkit を採用することにした。
  選択の背景、検討した代替案、トレードオフをまとめた ADR を
  docs/adr/ ディレクトリに Markdown 形式で作成して。
```

26.8 学習・キャッチアップ

26.8.1 既存コードベースの理解

引き継いだプロジェクトや大規模なコードベースを理解するのに Claude Code は特に役立ちます。

```
# プロジェクト全体の概要を把握する
claude -p "このプロジェクトの構成を説明して。主要なモジュールと、それぞれの役割を教えてほしい"

# 特定の機能の動作を理解する
claude -p "ユーザー認証の処理がどこで行われているか追って、処理フローを図解してほしい"
```

対話モードで深掘りするのも効果的です。

```
> この決済処理のコードを読んで、全体の流れを説明して

> (説明を受けた後) エラーが起きたとき、どの部分でトランザクションがロールバックされる?

> src/services/payment/rollback.ts の実装はどうなっている?
```

26.8.2 新しい技術の学習

新しい言語やフレームワークを学ぶとき、自分のプロジェクトに絡めて質問すると理解が深まります。

> GraphQL を始めて導入しようとしている。
現在 REST API を使っているが、このプロジェクトで GraphQL に移行する場合の
手順とベストプラクティスを教えてほしい。
特に認証周りの実装方法を詳しく教えてほしい。

> Rust を勉強中です。Python で書いた以下の関数を Rust で書き直してほしい。
コードの差分とともに、Rust ならではの考え方（所有権、借用など）を解説してほしい。

つまづきポイント: Claude は知識のカットオフ（学習データの終了日）があるため、最新バージョンの情報が正確でないことがあります。「このライブラリの最新の書き方を教えて」と聞くときは、バージョンを明示して「React 19 の use() フックの使い方を教えて」のように質問すると精度が上がります。公式ドキュメントと合わせて確認する習慣も大切です。

26.9 セキュリティ上の注意

Claude を開発に使う際は、以下の点に注意してください。

26.9.1 機密コードの取り扱い

Claude Web (claude.ai) や API にコードを送る際、以下のものが含まれていないか確認してください。

- **API キーや秘密鍵** — コード内にハードコードされた認証情報
- **個人情報** — ユーザーデータ、メールアドレス、氏名などが含まれるサンプルデータ
- **社内機密情報** — 公開されていない設計資料、顧客情報、ビジネスロジック

```
# .env ファイルは絶対に含めない
# 環境変数はプレースホルダーに置き換えてから共有する

# 悪い例（実際のキーが含まれる）
cat config.ts | claude -p "このコードをレビューして"

# 良い例（機密情報を除いたファイルのみ共有）
cat src/components/Button.tsx | claude -p "このコードをレビューして"
```

26.9.2 Claude Code のセキュリティ

Claude Code はローカルのファイルシステムに直接アクセスしますが、実行したコマンドや変更内容は会話履歴として Anthropic のサーバーに送られます。機密性の高いプロジェクトでは、利用するプランやデータ保持ポリシーを確認してください。



警告:

本番環境には直接実行しない

Claude Code に「本番環境のデータベースを更新して」などの指示を出す場合は注意が必要です。Claude が意図と異なる操作をする可能性があります。本番環境に影響する変更は、必ず内容を確認してから手動で実行してください。

26.9.3 コード生成の検証

Claude が生成したコードは、必ず以下を確認してから使用してください。

- **動作確認:** 実際にテストを実行して期待通りに動くか確認する
- **セキュリティ確認:** 特に認証・認可・入力バリデーション周りは慎重にレビューする
- **依存ライブラリ:** 生成コードが参照するライブラリが実際にインストールされているか確認する

26.10 実践 Tips

26.10.1 プロンプトの精度を上げるコツ

エンジニアリング用途では、以下のような指示の出し方が効果的です。

制約を明示する

```
> src/api/user.ts を修正して。  
ただし:  
- 関数のシグネチャは変えない  
- 外部ライブラリは追加しない  
- TypeScript の strict モードに準拠する
```

期待する出力形式を指定する

```
> このシステムのデータフローを説明して。  
Mermaid 図 (flowchart LR) の形式で出力してほしい。
```

段階的に進める

大きな変更を一度に頼むより、段階的に依頼する方がミスが少なくなります。

```
# Step 1: 現状把握  
> @src/services/ このサービス層の設計を分析して  
  
# Step 2: 方針確認  
> 依存性注入パターンを導入したい。どのように進めるか提案してほしい  
  
# Step 3: 実装  
> まず UserService から始めて、段階的にリファクタリングして
```

26.10.2 CLAUDE.md でチームの規約を共有する

チームで Claude Code を使う場合、CLAUDE.md にコーディング規約やアーキテクチャの方針を書いておくと、Claude の回答がプロジェクトに合ったものになります。

コーディング規約

命名規則

- 変数・関数: camelCase
- コンポーネント: PascalCase
- 定数: UPPER_SNAKE_CASE

アーキテクチャ方針

- ビジネスロジックは `src/services/` に集約する
- データベースアクセスは `src/repositories/` のみで行う
- コンポーネントはプレゼンテーション層に留め、ビジネスロジックを含めない

テスト方針

- ユニットテスト: Vitest を使う
- モックは `vi.fn()` を使い、外部依存はすべてモックにする
- テストファイルは実装ファイルと同じディレクトリに置く (`__tests__/`)

26.11 次のステップ

エンジニアリング活用をさらに深めるためのリソースです。

- [Claude Code をはじめよう](#) (ページ 129) — Claude Code のインストールから最初のコマンド実行まで
- [Claude Code 実践ワークフロー集](#) (ページ 147) — コードレビュー、バグ修正、テスト生成など実践的なワークフロー
- [IDE 統合](#) (ページ 159) — VS Code や JetBrains と Claude Code を組み合わせて使う方法
- [API 入門](#) (ページ 357) — CI/CD やスクリプトに Claude を組み込む方法
- [MCP ガイド](#) (ページ 345) — データベースや外部ツールを Claude Code に接続する方法

第27章 マーケティング活用ユースケース

「ブログを毎週書かないといけないのに、ネタ出しと執筆で時間が取られる」「SNS の投稿文を考えるだけで1時間かかる」「競合調査のレポートをまとめる作業が終わらない」

マーケターが日々抱えるこうした悩みに、Claude は強力な助けになります。このページでは、コンテンツ制作から市場分析まで、**そのままコピーして使えるプロンプト例**とともにマーケティング業務での活用シーンを紹介します。

27.1 マーケティングで Claude を使うメリット

Claude をマーケティング業務に取り入れると、大きく3つの恩恵を受けられます。

27.1.1 コンテンツ制作が圧倒的に速くなる

ブログ記事の下書き、SNS 投稿文、プレスリリース、メールマガジン——これらを一から書くとなると、ベテランのライターでも相当な時間がかかります。Claude を使えば、**アイデアの整理から下書きの生成まで一気に進められます**。もちろんそのまま使えるわけではありませんが、「ゼロから書く」のではなく「Claude が作った素材を磨く」という流れに変えるだけで、制作スピードは格段に上がります。

27.1.2 データ分析が専門知識なしでできる

競合分析や市場調査は、本来マーケティングリサーチの専門家が担う領域でした。Claude の Deep Research 機能（Pro プラン以上）を使えば、複数の情報源から情報を集めてレポート形式で整理してくれます。アナリストに頼まなくても、自分で深い調査ができます。

27.1.3 アイデア発想のパートナーになる

「新キャンペーンのコンセプトが浮かばない」「ターゲット層へのメッセージングをどう変えるか迷っている」——Claude は壁打ち相手として機能します。自分のアイデアを投げかけると、別の視点からフィードバックや派生アイデアを返してくれます。

つまづきポイント：「Claude が書いたコンテンツはそのまま使えるの？」と感じる方は多いです。AI が生成したコンテンツは下書きとして扱うのが基本です。ファクトチェック（事実確認）、ブランドボイスとの整合性確認、そして最終的な人の目によるレビューが欠かせません。Claude はコンテンツ制作の「アシスタント」であり、「完成品の自動生成機」ではありません。

27.2 コンテンツ制作 — ブログ・ニュースレター・プレスリリース

27.2.1 ブログ記事の下書きを作る

読者に読んでもらえるブログ記事を書くには、適切な構成と読みやすい文体が必要です。Claude にトピック、読者層、トーンを指定すれば、すぐに構成案と下書きを出してくれます。

プロンプト例: ブログ記事の構成案

以下の条件でブログ記事の構成案を作成してください。

- テーマ: 中小企業がInstagramを使って集客を増やす方法
- 読者: 飲食店や小売業を経営する 40 代のオーナー。SNS に詳しくない
- 記事の目的: 「まずやってみよう」という気持ちにさせる
- 文字数: 2,000 字程度
- トーン: 親しみやすく、専門用語を使わない

H2 見出し 5〜6 個と、各セクションの要点を箇条書きで出してください。

プロンプト例: 下書きを書いてもらう

上記の構成をもとに、「はじめに」と「1. まずはプロフィールを整えよう」のセクションの本文を書いてください。

文体の条件:

- 難しい横文字はなるべく避ける（使う場合は括弧で説明を追加）
- 読者の「あるある」な悩みから始めて共感を引き出す
- 具体的な数字や例を必ず入れる

つまづきポイント: 「書いてもらったけど、うちのブランドっぽくない」と感じたら、既存の人気記事を Claude に読ませて「このスタイルに合わせて」と指示するのが効果的です。数記事を参考として貼り付けるだけで、文体がぐっと近づきます。

27.2.2 プレスリリースを素早く作る

新商品発表や企業ニュースのプレスリリースも、型が決まっているだけに Claude が得意とする領域です。

プロンプト例: プレスリリースの下書き

以下の情報をもとに、配信用プレスリリースの下書きを作成してください。

【発表内容】

- 新製品: エコバッグ「GreenPack」シリーズ
- 発売日: 2025 年 4 月 15 日
- 価格: 3,980 円〜
- 特徴: 再生プラスチックを 100%使用、折りたたみ可能、10L 容量
- 発売の背景: プラスチック削減への取り組みとして開発

(次のページに続く)

(前のページからの続き)

【形式】

- 標準的なプレスリリース形式（見出し・リード・本文・会社概要）
- 文字数：800 字程度
- 配信先：一般メディア・消費者向けメディア

27.2.3 ニュースレターの件名と本文

読者が開封してくれるニュースレターには、引きのある件名と読み進めたくなる冒頭文が欠かせません。

プロンプト例: 件名のバリエーション

以下のニュースレターの件名を 10 パターン作成してください。

内容：春の新商品コレクション発表（アパレルブランド向け）

読者層：30 代女性、ファッションに関心が高い

バリエーション：

- 好奇心を刺激するもの：3 パターン
- 限定感・緊急感を出すもの：3 パターン
- シンプルで親しみやすいもの：4 パターン

27.3 SNS 運用 — 投稿文・ハッシュタグ・スケジュール

SNS 運用では「量」と「一貫性」が求められます。毎日の投稿文を考えるだけで消耗してしまう、という担当者は多いですが、Claude を使えば複数のプラットフォーム向けに一気に投稿文を用意できます。

27.3.1 プラットフォーム別の投稿文を作る**プロンプト例: Instagram / X / LinkedIn 向けに書き分ける**

以下の内容を、SNS プラットフォームごとに最適な投稿文に書き分けてください。

【投稿内容】

新商品「オーガニックドリップコーヒー」の発売告知。

特徴：エチオピア産豆、1 杯あたり 200 円、農家直輸入で公平な取引を実現。

【各プラットフォームの条件】

- Instagram: 親しみやすいトーン。絵文字を使う。ハッシュタグを 10 個追加
- X (旧 Twitter): 140 文字以内。インパクトのある書き出し
- LinkedIn: ビジネス寄りのトーン。ブランドの社会的価値（フェアトレード）を強調。300 字程度

つまづきポイント: 「ハッシュタグが多すぎる」「投稿のトーンが自社っぽくない」という場合は、条件に「当社のブランドガイドライン: ○○」を追加したり、既存の人気投稿を参考として見せたりすると精度が上がります。

27.3.2 1 週間分の投稿スケジュールを作る

プロンプト例: 投稿スケジュール案

Instagram アカунツ用に、1 週間分の投稿スケジュール案を作成してください。

【アカウント情報】

- 業種: 地方の焙煎コーヒー専門店
- フォロワー層: 20〜40 代のコーヒー好き
- 投稿頻度: 週 5 回 (月〜金)

【各投稿の形式】

- 曜日: 投稿タイプの提案 (商品紹介・情報系・エンタメ系 etc.)
- キャプションの方向性 (2〜3 文で)
- ハッシュタグカテゴリ (3 つ程度)
- 最適な投稿時間帯

27.4 SEO — キーワードリサーチと構成案

検索エンジンで上位表示を狙うコンテンツを作るには、適切なキーワード選定と構成が重要です。Claude は SEO の専門ツールではありませんが、**コンテンツの戦略立案と構成作成**では大いに役立ちます。

27.4.1 関連キーワードを洗い出す

プロンプト例: キーワード展開

「ホームジム 作り方」というメインキーワードについて、SEO コンテンツ戦略を立てるためのキーワード分析をしてください。

1. 検索意図別の関連キーワードを 20 個リストアップ (購入検討層・情報収集層・初心者層に分類)
2. 「ホームジム」に関連する「よくある質問 (People Also Ask)」を 10 個想定
3. これらのキーワードを使って、「ピラーページ + クラスターページ」構造で作るとしたら、どんな記事構成が考えられるか提案してください

27.4.2 メタディスクリプションを生成する

プロンプト例: メタディスクリプション

以下のブログ記事のメタディスクリプションを 3 パターン作成してください。

記事タイトル: 「ホームジムを 5 万円で作る完全ガイド【2025 年版】」

ターゲット読者: 運動不足を感じている 30〜40 代の会社員

条件:

- 120〜160 文字以内

(次のページに続く)

(前のページからの続き)

- CTR（クリック率）を高める表現を意識
- 主要キーワード「ホームジム」「5万円」を含める
- 各パターンで訴求角度を変える（価格重視・手軽さ重視・結果重視）

つまづきポイント: Claude は現在のリアルタイム検索データにアクセスできないため、「このキーワードの月間検索ボリュームは？」といった質問には答えられません。検索ボリュームの確認は Google Keyword Planner や Ahrefs などの専用ツールを使い、Claude はあくまでコンテンツ設計や文章生成に活用するのが正しい使い方です。

27.5 市場分析・競合調査 — Deep Research で深掘りする

Claude の **Deep Research 機能**（Pro プラン以上）を使えば、複数の情報源を横断した市場調査レポートを自動生成できます。これまで数時間かかっていたリサーチ作業が、大幅に短縮されます。

27.5.1 市場調査レポートを依頼する

プロンプト例: 市場調査（Deep Research 使用）

日本のサブスクリプションボックス市場について調査レポートを作成してください。

以下の項目を含めてください:

1. 市場規模と成長率（過去3年 + 今後の予測）
2. 主要プレイヤーと各社の戦略
3. 消費者ニーズと購買動機
4. 市場参入する際の課題と機会
5. 海外（特に米国・欧州）との比較

レポート形式: エグゼクティブサマリー付きの構成化されたレポート

分量: 2,000〜3,000 字程度

※ Deep Research を使用するには、チャット画面の入力欄左にある「Deep Research」をオンにしてから送信します。

27.5.2 競合の SWOT 分析

プロンプト例: 競合 SWOT 分析

以下の企業について SWOT 分析を実施してください。

対象企業: [競合他社名]

分析の目的: 自社の差別化ポイントを明確にし、来期のマーケティング戦略に活かす

SWOT の各項目について:

- 強み（Strengths）: 5 点
- 弱み（Weaknesses）: 5 点

(次のページに続く)

(前のページからの続き)

- 機会 (Opportunities) : 5 点
- 脅威 (Threats) : 5 点

最後に「自社が差別化できる領域」として、3つの具体的な提言をまとめてください。

つまづきポイント: Claude の学習データには時間的な限界 (カットオフ) があります。最新の競合情報や直近のトレンドについては、Deep Research を使っても完全に最新とは限りません。重要な意思決定に使う際は、必ず公式サイトや業界レポートなど一次情報で内容を検証してください。

27.6 広告コピー — A/B テスト用バリエーションを一気に作る

広告コピーの A/B テストは、成果を改善するうえで欠かせない取り組みです。しかし、テスト用のバリエーションを複数作るのは時間がかかります。Claude を使えば、異なるアプローチで複数のコピーを一度に生成できます。

27.6.1 A/B テスト用コピーを生成する

プロンプト例: 広告コピーのバリエーション

以下の商品の Web 広告コピーを、A/B テスト用に 5 パターン作成してください。

【商品情報】

- 商品名: オンライン英会話「スピードイングリッシュ」
- 価格: 月額 4,980 円
- 特徴: AI による発音矯正、1 回 25 分、スマホで完結
- ターゲット: 昇進・転職を考えている 30 代ビジネスパーソン

【各パターンの訴求軸】

1. 成果・実績重視 (「〇〇が変わる」系)
2. 手軽さ・継続しやすさ重視
3. 不安解消・背中を押す系
4. 競合比較訴求
5. 限定感・今すぐ感を出す系

各パターン: 見出し (30 字以内) + 説明文 (80 字以内) + CTA (15 字以内)

27.6.2 ターゲット別メッセージを最適化する

同じ商品でも、ターゲットによって響くメッセージは異なります。

プロンプト例: ターゲット別のメッセージ最適化

「健康診断で血糖値が高いと言われた人向けの低糖質食品」について、ターゲットセグメント別のキャッチコピーを作成してください。

セグメント：

1. 「医師に言われて仕方なく」始めようとしているシニア層
2. 「健康的にダイエットしたい」30代ビジネスパーソン
3. 「家族のために食事を変えたい」主婦・主夫層

各セグメントについて：

- そのセグメントの主な不安・動機
- キャッチコピー（20字以内）：3パターン
- LP のファーストビュー用の説明文（100字以内）

27.7 メールマーケティング — セグメント別の文面を効率化

メールマーケティングで成果を出すには、読者セグメントに合わせたパーソナライズが重要です。しかし、セグメントごとに別々のメールを書くのは大変な作業です。

27.7.1 セグメント別メール文面を作る

プロンプト例: セグメント別メール

EC サイトの既存顧客向けに、「夏セール」案内メールをセグメント別に作成してください。

【セグメントと条件】

セグメント 1: 直近 3 ヶ月以内に購入した顧客（アクティブ層）

- 目的: リピート購入の促進
- トーン: 親しみやすく、特別感を演出

セグメント 2: 6 ヶ月以上購入がない顧客（休眠層）

- 目的: 再購入のきっかけ作り
- トーン: 「お久しぶり」感を出しつつ、お得な情報で引き寄せる

各セールメールに含める要素：

- 件名（A/B テスト用に 2 パターン）
- 本文（300 字程度）
- 明確な CTA（クリックを促す一文）

つまづきポイント: メール文面で個人名差し込み（「〇〇様、こんにちは」）などを使う場合、Claude が生成した文面にそのままプレースホルダーを手動で追加する必要があります。メール配信ツール（Mailchimp、Salesforce Marketing Cloud 等）との連携方法は別途確認してください。

27.7.2 件名のバリエーションを大量生成する

メール開封率は件名で大きく変わります。テストしたい件名のバリエーションを一度にたくさん作ってもらいましょう。

プロンプト例: 件名のバリエーション大量生成

「初回限定 50%オフキャンペーン」メールの件名を 20 パターン作成してください。

対象: 美容系 D2C ブランドの新規登録ユーザー (20~30 代女性)

以下のアプローチを混在させてください:

- 数字・割引率を前面に出すもの: 5 パターン
- 好奇心・謎めかしたもの: 5 パターン
- 限定性・期限を強調するもの: 5 パターン
- パーソナル・親しみやすいもの: 5 パターン

27.8 レポート作成 — KPI レポートと経営層向けサマリー

月次のマーケティングレポートや経営層向けのサマリー作成も、Claude が助けてくれます。数字やデータをそのまま貼り付ければ、わかりやすいまとめを作ってくれます。

27.8.1 KPI レポートのサマリーを作る

プロンプト例: マーケティング KPI レポート

以下の月次マーケティングデータをもとに、経営会議用のサマリーレポートを作成してください。

【5 月の実績データ】

- Web サイト訪問者数: 42,500 (前月比 +8%)
- 問い合わせ件数: 380 件 (前月比 +12%)
- メール開封率: 24.3% (業界平均: 22%)
- SNS フォロワー増加数: Instagram +1,200、X +350
- リスティング広告: 表示回数 180 万、クリック率 3.2%、CV 数 95 件、CPA 12,000 円

【作成条件】

- 読者: マーケティングに詳しくない経営陣
- 形式: 箇条書き中心で、見やすいサマリー
- 分量: A4 1 枚以内に収まる量
- 強調すべき成果と課題をそれぞれ 3 点ずつ
- 来月の推奨アクションを 3 点

27.8.2 キャンペーン結果の振り返りレポート

プロンプト例: キャンペーン結果分析

以下のキャンペーンの結果を振り返り、今後の改善点を整理するレポートを作成してください。

【キャンペーン概要】

- 施策: 新商品発売記念 Instagram 広告キャンペーン (3月1日～31日)
- 予算: 50万円
- ターゲット: 25～35歳女性

【結果データ】

- リーチ数: 280万人
- エンゲージメント率: 1.8%
- LP への流入: 4,200件
- CV (購入): 126件
- 売上: 756,000円 (商品単価: 6,000円)
- ROAS (広告費用対効果): 1.51

【作成してほしいこと】

1. 数字の評価 (良かった点・悪かった点)
2. ROAS が低い原因の仮説 (3～5点)
3. 次回キャンペーンへの改善提案 (具体的なアクション付き)

27.9 AI 生成コンテンツを使う際の注意点

Claude はマーケティング業務の強力な助けになりますが、使う際にはいくつかの点を意識してください。

27.9.1 ファクトチェックは必ず行う

Claude が生成した文章に含まれる数字、統計、固有名詞などは、必ず一次情報で確認してください。AI は「もっともらしい情報」を生成することがあり、細かい数値などで誤りが混入することがあります。特に**外部向けに公開するコンテンツ** (プレスリリース、ブログ記事等) は重点的にチェックが必要です。

27.9.2 ブランドボイスとの整合性を確認する

Claude が生成した文章は、あくまで汎用的なものです。自社のブランドガイドライン (トーン、禁止ワード、表現スタイルなど) と照らし合わせて、必ず編集・加筆してください。毎回「自社のブランドガイドライン」を Claude に読ませることで、生成の精度を高めることができます。

27.9.3 著作権とオリジナリティについて

Claude が生成したコンテンツは、他の既存コンテンツと完全に一致することはほぼありませんが、構成や表現の類似が生じる可能性があります。重要なコンテンツを公開する前に、コピーチェックツール (CopyScape 等) を使って確認する習慣をつけましょう。また、Claude の出力をそのままコピーして公開するのではなく、**自社の視点・経験・データを加えてオリジナルのコンテンツに仕上げる**ことが、SEO 上も読者への価値提供の観点からも重要です。

つまづきポイント: 「Claude に頼りすぎてコンテンツが均質化した」という悩みを聞くことがあります。Claude は補助ツールとして位置づけ、企画立案や構成作りには使いつつ、核心となるメッセージや独自の視点は人間が加えることで、差別化されたコンテンツが生まれます。

27.10 次のステップ

Claude でマーケティング業務をさらに効率化するために、以下の記事もあわせてご覧ください。

- [Claude Web \(claude.ai\) の使い方](#) (ページ 89) — プロンプトを入力してコンテンツを生成する基本操作を学ぶ
- [Deep Research の使い方](#) (ページ 105) — 市場調査・競合調査に役立つ Deep Research 機能の詳細
- [プロンプトの基本原則](#) (ページ 239) — 「そのまま使えるプロンプト例」を自分流にアレンジするためのスキルを身につける
- [Projects の活用法](#) (ページ 89) — ブランドガイドラインや商品情報を登録して、毎回の出力品質を上げる

第28章 プロンプトの基本原則

「Claude に頼んでみたけど、思ったような回答が返ってこない」「毎回違う品質の答えが来て、安定しない」——そんな経験はありませんか？

その原因の多くは、Claude の能力ではなく **指示の書き方（プロンプト）** にあります。同じ Claude でも、指示の仕方次第で回答の質は大きく変わります。このページでは、あらゆる Claude 製品で使える「よい指示の書き方」の基本原則を学びます。

テンプレートを暗記する必要はありません。3つの原則を理解すれば、自分のあらゆるタスクに応用できるようになります。

28.1 なぜ「指示の書き方」が重要なのか

同じ質問でも、聞き方によって返ってくる答えはまったく違います。

次の2つの指示を比べてみてください。

指示 A（曖昧な例）：

メールを書いて

指示 B（明確な例）：

以下の状況に合わせた取引先への依頼メールを書いてください。

状況：来週の打ち合わせ日程を変更したい

相手：3年間取引のあるメーカーの営業担当者

変更理由：社内で急な会議が入った

希望の新日程：翌週の月曜日から火曜日午後

トーン：丁寧かつ簡潔に。謝罪の言葉も一言添えてください

指示 A に対して Claude は「どのメールか」「誰宛か」「何の目的か」がわからないため、汎用的で使えないメールを書くか、逆質問してきます。指示 B なら、すぐに使える質の高いメールが返ってきます。

つまづきポイント：「Claude は賢いんだから、細かく言わなくてもわかるはず」と思いがちですが、Claude は **あなたの頭の中にある文脈を知りません**。人間の同僚に仕事を頼むときと同じように、「背景」「ゴール」「条件」を丁寧に伝えることが大切です。

この差を生み出すのが、これから説明する3つの基本原則です。

28.2 原則 1: 明確に伝える

最初の原則は「曖昧さをなくすこと」です。

Claude は指示に書かれていないことは推測で補います。その推測があなたの意図と一致しているとは限りません。指示が曖昧なほど、Claude の解釈とあなたの期待がズレる可能性が高くなります。

28.2.1 曖昧な言葉に気をつける

「いい感じに」「適切に」「わかりやすく」といった表現は、人によって解釈が違います。何をもって「いい感じ」なのかを具体的に言い換えましょう。

ビフォーアフター 1: 文章の修正依頼

× これをいい感じに直して

[長い文章]

✓ この文章を以下の基準でリライトしてください：

- 文を短くする（1 文は 60 文字以内）
- 受動態を能動態に変える
- 結論を最初の段落に持ってくる

[長い文章]

ビフォーアフター 2: 資料の要約依頼

× この PDF を要約して

[PDF の内容]

✓ この資料の要約を作成してください。

要件：

- 300 文字以内
- 箇条書き形式（3～5 点）
- 経営層への報告に使うので専門用語は避ける
- 最重要の数字（売上、コスト等）は必ず含める

[PDF の内容]

ビフォーアフター 3: アイデア出し依頼

× 新商品のアイデアを考えて

✓ 以下の条件で新商品のアイデアを 10 個出してください。

前提：

(次のページに続く)

(前のページからの続き)

- 対象: 40〜60 代のオフィスワーカー
- 価格帯: 3,000〜8,000 円 (ギフトに最適な価格)
- カテゴリ: 文具・デスク用品
- 競合との差別化: エコ素材を使った日本製であること

28.2.2 「何をしてほしいか」をはっきり書く

動詞を明確にするだけで、回答の質が上がります。

曖昧な指示	明確な指示
「この文章どう？」	「この文章の論理的な矛盾点を指摘してください」
「競合について教えて」	「A 社・B 社・C 社の料金プランを比較表にしてください」
「このメール見て」	「このメールの返信文を 3 パターン書いてください (丁寧・簡潔・強調版)」

つまづきポイント: 「明確に書かなければいけない」と意識しすぎると、プロンプトが長くなりすぎることがあります。最初は箇条書きで「やってほしいこと」「やってほしくないこと」「出力形式」の 3 点だけ書く習慣から始めましょう。

28.3 原則 2: 具体的に伝える

「明確さ」と似ていますが、「具体性」は **数値・形式・基準を使って細部を指定すること** です。「何を・どのように・どの程度」を指定するほど、期待に近い出力が得られます。

28.3.1 「何を」——出力の内容を指定する

何を含めてほしいか、何を含めなくていいかを書きます。

ビフォーアフター 4: 企画書の作成

× 新サービスの企画書を作って

✓ 以下の構成で新サービスの企画書の本文を作成してください。

必ず含めるもの:

- サービスの概要 (100 文字以内)
- ターゲットユーザーの定義
- 競合との差別化 3 点
- 初年度の収益見通し (数字のプレースホルダーで OK)

含めなくていいもの:

- 詳細な技術仕様
- 財務諸表

(次のページに続く)

(前のページからの続き)

サービス情報: [サービスの情報をここに書く]

28.3.2 「どのように」——出力の形式を指定する

文章か箇条書きか、表か段落か。形式を指定すると、そのまま使えるアウトプットが返ってきます。

✓ 会議の議事録を以下のフォーマットで作成してください:

[日付] [会議名]

参加者:

決定事項:(箇条書き)

課題・検討事項:(箇条書き)

次回アクション:(担当者・期限付き箇条書き)

以下の会議メモを整理してください:

[会議メモをここに貼る]

28.3.3 「どの程度」——量・深さを指定する

文字数、項目数、詳細度を指定すると、過不足のない回答が返ってきます。

ビフォーアフター 5: 説明の依頼

× AI について教えて

✓ AI について、以下の条件で説明してください:

- 対象: IT 知識のない経営者 (50 代)
- 文字数: 500 文字程度
- 専門用語は使わず、身近なたとえを使う
- 「自社のビジネスにどう使えるか」という視点を入れる

ビフォーアフター 6: タスク分解の依頼

× このプロジェクトの計画を立てて

✓ 以下のプロジェクトを週次タスクに分解してください。

プロジェクト: 社内ナレッジベースの構築

期間: 3 ヶ月

担当: 私 1 人 (兼務。1 日 2 時間程度)

ゴール: 部署の業務マニュアルをすべてデジタル化して Wiki に格納する

出力形式:

(次のページに続く)

(前のページからの続き)

- 第 1 ヶ月・第 2 ヶ月・第 3 ヶ月に分けて
- 各週に 2〜3 個のタスク
- 各タスクの目安時間（時間数）も記載

つまづきポイント: 具体的に書こうとするあまり、「完璧なプロンプト」を作ることに時間をかけすぎる人がいます。まず大まかに依頼して、返ってきた回答を見て「もっとこうしてほしい」と続きのメッセージで調整するほうが、多くの場合は効率的です。

28.4 原則 3: 文脈を提供する

3 つ目の原則は「背景・目的・制約を伝えること」です。

Claude は指示を文字どおりに実行しようとします。しかし、同じ作業でも「誰のために・何の目的で・どんな制約のもとで」によって、最適なアウトプットは変わります。この情報（文脈）を提供することで、Claude はあなたの状況に合った回答をしてくれます。

28.4.1 「なぜ」を伝える——目的を共有する

目的を伝えると、Claude が「本当に役立つこと」を判断できます。

ビフォーアフター 7: 文章チェックの依頼

× この提案書の文章をチェックして

[提案書]

✓ この提案書の文章をチェックしてください。

目的: 来週の経営会議で、予算 300 万円の承認を得るために使います。

読者: 技術に詳しくない 50 代の役員 3 名

重点的にチェックしてほしいこと:

- 投資対効果が明確に伝わるか
- 反論されそうな点を先回りして答えているか
- 全体の論旨に矛盾がないか

[提案書]

28.4.2 「誰が」を伝える——立場・役割を共有する

あなたが誰で、どんな立場から依頼しているかを伝えると、回答が状況に合ったものになります。

✓ 私は中小企業（従業員 30 名）の経理担当者です。

経理ソフトの乗り換えを検討していますが、

現在の弊社の規模・業務量に合ったソフトウェアの選び方を

教えてください（クラウド型のみ、月額予算は 3 万円以内）。

28.4.3 Role / Task / Format のフレームワーク

文脈を整理するためのシンプルなフレームワークとして、**Role（役割） / Task（タスク） / Format（形式）**があります。

要素	内容	例
Role	Claude に演じてほしい役割	「あなたはベテランの人事担当者です」
Task	やってほしいこと	「以下の求人票を添削してください」
Format	出力の形式	「修正点を箇条書きで、各点に理由もつけて」

この 3 要素を意識するだけで、プロンプトの構造が自然と整います。

実際の例（求人票の改善依頼）：

あなたは採用広報の専門家で、求人票のライティングに 10 年以上の経験があります。

(Role)

以下の求人票を添削してください。

(Task)

以下の観点で評価してください：

1. 応募者にとって魅力的に見えるか
2. 会社の文化・価値観が伝わるか
3. 求める人物像が具体的か

修正案は元の文章の横に【修正後】として示してください。

(Format)

[求人票をここに貼る]

ビフォーアフター 8: レポート作成の依頼

× 市場調査のレポートを書いて

✓ あなたはマーケティングコンサルタントです。

私は国内向けのオーガニック食品ブランドを立ち上げようとしています。

以下の調査結果をもとに、投資家向けの市場調査レポートを作成してください。

調査データ：

- 国内オーガニック食品市場規模：2023 年 約 2,300 億円
- 年間成長率：約 7%
- 主要購買層：30～50 代の女性（特に子育て世代）

レポートの要件：

- A4・2 ページ相当（約 1,200 文字）
- 「市場機会」「競合状況」「参入戦略」の 3 章構成

(次のページに続く)

(前のページからの続き)

- 投資家向けなので数字とリスクを明確に記載

[追加の調査データがあればここに貼る]

つまづきポイント:「文脈を伝える」のが大事と言われても、どこまで書けばいいか迷いますよね。目安は「初めてこの仕事を頼む外部のフリーランサーに送る依頼メール」のイメージです。読んだ人が「何を・なぜ・どのくらいの品質で」やればいいかがわかるレベルを目指しましょう。

28.5 3 原則をまとめて使う

3 つの原則は、それぞれ独立して使えますが、組み合わせると最大の効果を発揮します。

次の例は、日常的なビジネスタスク（週次報告書の作成）に 3 原則をすべて適用したものです。

改善前（原則なし）：

週次報告書を書いて

改善後（3 原則を適用）：

以下の情報をもとに、今週の週次報告書を作成してください。

【目的と読者（文脈）】

- 読者：直属の上長（部門長）
- 目的：業務の進捗確認と、来週の方針決定のための情報共有
- 私の役職：営業部 係長

【今週の主な活動（情報）】

- 月曜：A 社への提案書を完成、メールで送付
- 火曜～水曜：B 社との商談（3 回）。先方の予算感が判明
- 木曜：新規リードへのフォローアップコール 8 件
- 金曜：月次の売上集計作業

【報告書の要件（明確・具体的）】

- 形式：「今週の実績」「来週の予定」「相談事項」の 3 セクション
- 分量：各セクション 200 文字以内
- 数字は具体的に（訪問件数、商談件数など）
- 読みやすいよう箇条書きを使う

このように 3 原則を意識するだけで、そのまま使える週次報告書が返ってきます。

28.6 実践ワークショップ

読んだだけでは身につけません。実際に手を動かして、自分のプロンプトを改善してみましょう。

28.6.1 問題 1: 曖昧な指示を明確にする

次の指示を、3 原則を使ってより明確に書き直してください。

元の指示:

競合調査してほしい

ヒント:

- どの競合？（業界・会社名）
- 何を調べる？（価格？機能？マーケティング？）
- どんな形式で欲しい？（表？レポート？）
- 何のために使う？（社内共有？プレゼン？）

改善例:

国内のクラウド型経費精算ソフトの競合調査をしてください。

調査対象：楽楽精算・マネーフォワード経費・freee 経費 の 3 社

調査項目：

- 月額料金（ユーザー数 30 名の場合）
- 主な機能の有無（スマホ申請・承認フロー・会計ソフト連携）
- 対象規模（中小企業向け/大企業向け）

出力形式：比較表（行=製品名、列=調査項目）

用途：来月の社内会議で導入製品を決定するための資料

28.6.2 問題 2: 文脈を追加する

次の指示に「文脈」を追加して、よりの確な回答が得られるように改善してください。

元の指示:

英語でメールを書いて

ヒント:

- 誰に宛てたメール？（取引先？社内の海外拠点？）
- 何の目的で？（依頼？報告？謝罪？）
- どのくらいのフォーマル度？
- 日本語での下書きはある？

改善例:

以下の日本語の下書きを、英語のビジネスメールに翻訳してください。

相手: アメリカのサプライヤー (長年の取引先)

目的: 来月の発注数量の変更を伝える

トーン: 丁寧かつビジネスライク。過度に堅苦しくない関係

言語: アメリカ英語

日本語下書き:

「来月 3 月の発注数量について、当初の 1,000 個から 1,200 個に変更させてください。
需要予測の上方修正があったためです。ご対応可能かご確認をお願いします。」

28.6.3 問題 3: 自分のプロンプトを改善する

あなたが最近 Claude に送ったプロンプト (または普段よく使う指示) を 1 つ取り上げてください。

以下のチェックリストで見直してみましょう:

チェックリスト:

- ☐ **明確さ:** 動詞が具体的か? (「見て」より「チェックして修正案を出して」)
- ☐ **具体性:** 文字数・件数・形式を指定しているか?
- ☐ **文脈:** 目的・読者・制約を伝えているか?
- ☐ **Role:** 必要なら Claude に役割を与えているか?
- ☐ **Format:** 出力形式 (箇条書き・表・段落など) を指定しているか?

1 つでも「いいえ」と答えた項目があれば、その情報を追加してみてください。多くの場合、回答の質が目に見えて改善します。

**注釈:**

ワークショップは読むだけでは効果が半減します。実際に Claude を開いて、改善したプロンプトを送ってみましょう。「前のプロンプトとどう違うか」を比べることで、原則の効果を体感できます。

28.7 よくある質問**Q: プロンプトが長くなりすぎて不安です。短くても大丈夫ですか?**

A: 短いプロンプトでも大丈夫な場面はたくさんあります。「このメールの要約を 3 行で」のような単純な依頼には、短い指示で十分です。長くする必要があるのは、「品質の基準が厳しい」「誤解されると困る」「専門的な文脈が必要」な場合です。

Q: 毎回詳しく書くのが面倒です。何かいい方法がありますか?

A: よく使うプロンプトのテンプレートを作っておくと便利です。たとえば「議事録要約テンプレート」を一度作って保存しておけば、次回からは会議メモを貼り付けるだけで使えます。Claude の Projects 機能を使えば、よく使う指示をシステム設定として保存することもできます。

Q: Claude が思った答えを出してくれないとき、どうすればいいですか？

A: 同じ会話のなかで「もっと短く」「もっと具体的な数字を入れて」「この部分だけ書き直して」と続きのメッセージで修正依頼ができます。1 回で完璧を目指さず、対話しながら改善していくのが実は一番効率的です。

28.8 次のステップ

3 つの基本原則を理解したら、次は「特定の状況で繰り返し使えるパターン（型）」を学びましょう。Chain of Thought（段階的思考）、Few-shot（具体例の提示）、XML タグ構造化など、より高度なテクニックを紹介しています。

- **プロンプトパターン集** (ページ 249) — Chain of Thought、Few-shot、XML タグなど、7 つの再利用可能なパターンを解説しています。本記事の続編です。

第29章 プロンプトパターン集

前編の「プロンプトの基本原則」では、明確さ・具体性・文脈提供という3つの基本を学びました。この記事では一歩進んで、特定の状況で繰り返し使える「プロンプトパターン」を紹介します。

パターンとは「型」のことです。しかし、ここで大切なのは **テンプレートをコピペすることではなく、「なぜその型が機能するのか」を理解すること** です。考え方を理解すれば、自分のタスクに合わせて自由に組み合わせ・応用できるようになります。

29.1 パターンの読み方

この記事では、各パターンを以下の統一フォーマットで解説します。

項目	内容
いつ使う	このパターンが特に有効な場面
構造	プロンプトの組み立て方
例	ビジネス寄り・開発寄りの具体的なプロンプト例
注意点	よくある失敗とその対策

7つのパターンを紹介しますが、最終セクションで「組み合わせ」の実践例も載せています。まず一通り読んだ後、自分のタスクに近い組み合わせを試してみてください。

29.2 パターン 1: Chain of Thought (段階的思考)

29.2.1 いつ使う

- 数値計算や論理推論が必要なとき
- 複数の要素を比較・評価するとき
- 答えよりも「なぜその答えなのか」のプロセスが重要なとき

Chain of Thought (CoT) は、Claude に「考える手順を明示させる」パターンです。人間でも、難しい問題を「まず〇〇を確認して、次に〇〇を検討して...」と声に出しながら考えると整理できますが、それと同じことを Claude に促します。

29.2.2 構造

最もシンプルな方法は、プロンプトの末尾に「ステップごとに考えてください」という一文を加えることです。

[タスクの説明]

ステップごとに考えを整理しながら答えてください。

より明示的にする場合は、考えるべき手順を自分で指定します。

[タスクの説明]

以下の順で検討してください：

1. まず〇〇を確認する
2. 次に〇〇を評価する
3. 最後に結論を出す

29.2.3 例

ビジネス寄り（予算の妥当性検討）：

以下の新規事業の予算案を評価してください。

事業名：社内 AI 活用支援サービス

初期投資：500 万円

年間運用費：120 万円

想定収益：年間 200 万円（3 年目から）

ステップごとに考えを整理しながら評価してください：

1. 投資回収期間の計算
2. リスク要因の洗い出し
3. 類似事業との比較
4. 総合的な判断

開発寄り（コードのバグ調査）：

以下の Python 関数が期待通りに動作しない原因を特定してください。

```
```python
def calculate_average(numbers):
 total = 0
 for n in numbers:
 total += n
 return total / len(numbers)
```
```

(次のページに続く)

(前のページからの続き)

入力: [1, 2, 3]

期待する出力: 2.0

実際の出力: 2.0（一見正しいが、空リストを渡すとクラッシュする）

考えられるエッジケースをすべて洗い出してから、修正案を提示してください。

29.2.4 注意点

よくある失敗: 「ステップごとに」と言うだけで満足してしまう

「ステップごとに考えてください」という指示は万能ではありません。複雑なタスクでは、考えるべき観点を自分で指定したほうが精度が上がります。「何を考えてほしいか」をあらかじめ整理しておくことが、このパターンの真の使い方です。

29.3 パターン 2: Few-shot（具体例の提示）

29.3.1 いつ使う

- 出力のフォーマットを厳密に指定したいとき
- 文体や表現のトーンを揃えたいとき
- 「言葉では説明しにくいが見ればわかる」品質基準があるとき

Few-shot（少数ショット）は、「こういう入力にはこういう出力を」という具体例を複数示すことで、Claude が期待するパターンを学習するアプローチです。料理のレシピと同様に、材料（入力）と完成品（出力）を見せれば、同じ調理法で新しい料理を作ってもらえます。

29.3.2 構造

以下の例を参考に、同じ形式で [タスク] を実施してください。

---例 1---

入力: [入力例 1]

出力: [出力例 1]

---例 2---

入力: [入力例 2]

出力: [出力例 2]

---ここから本番---

入力: [実際の入力]

出力:

例の数は 2～5 個が適切です。1 個だと「たまたまその形式になった」と判断される可能性があります。5 個を超えると余分なトークンを消費する割に改善幅が小さくなります。

29.3.3 例

ビジネス寄り（会議メモの要約）：

以下の形式で会議メモを要約してください。

---例 1---

入力：

「本日の会議では、Q3 の販売実績について議論しました。田中部長から、目標比 120%達成との報告がありました。ただし地方の数字が弱く、来期の施策として地方営業強化を検討することになりました。」

出力：

【決定事項】 地方営業強化施策を来期に向けて検討する
【共有事項】 Q3 販売実績：目標比 120%達成。地方拠点が課題
【次のアクション】 なし

---例 2---

入力：

「エンジニアチームから、新機能の開発が来週完了予定との報告がありました。リリースにあたって QA チームとの調整が必要で、山田さんが窓口になることが決まりました。リリース日は未定です。」

出力：

【決定事項】 QA 調整の窓口：山田さん
【共有事項】 新機能開発、来週完了予定。リリース日は調整中
【次のアクション】 山田さん：QA チームへの連絡

---ここから本番---

入力：

「本日は予算会議でした。来期の広告費について議論し、前期比 10%増の 1,100 万円で仮承認されました。ただし役員会での最終承認が必要です。鈴木さんが来週の役員会でプレゼンを担当します。」

出力：

開発寄り（コミットメッセージの生成）：

以下の形式でコミットメッセージを生成してください。

---例 1---

変更内容： ユーザーが 404 エラーのページで「ホームに戻る」ボタンを押せなかった

出力： fix: 404 ページの CTA ボタンのクリックイベントが未登録だった問題を修正

---例 2---

変更内容： ユーザープロフィール画面に生年月日フィールドを追加した

出力： feat: プロフィール画面に生年月日の入力・表示機能を追加

(次のページに続く)

(前のページからの続き)

---例 3---

変更内容: API のレスポンス速度が遅かったのでキャッシュを入れた

出力: perf: ユーザー一覧 API に Redis キャッシュを導入し応答速度を改善

---ここから本番---

変更内容: パスワードリセット機能が特殊文字を含むメールアドレスでエラーになっていた

出力:

29.3.4 注意点

よくある失敗: 例が少なすぎる、または偏っている

例が 1~2 個しかない場合や、似たパターンの例ばかり提示すると、Claude がパターンを正しく一般化できません。バリエーションのある例を 3~5 個用意しましょう。また例が長すぎるとプロンプト全体が肥大化するので、できるだけ簡潔な例を選びます。

29.4 パターン 3: XML タグ構造化

29.4.1 いつ使う

- プロンプトが長くなってきた (300 文字以上が目安)
- 「指示」「背景情報」「入力データ」が混在している
- Claude に特定の形式で出力させたい

XML タグとは `<instructions>` や `<context>` のような山かっこで囲んだ目印のことです (実際に XML を知らなくても問題ありません)。Claude はこのタグを「区切り記号」として認識し、プロンプトの各部分の役割を把握しやすくなります。

Claude は特に XML タグへの対応力が高く、タグを使うだけで回答の精度が目に見えて改善することがあります。

29.4.2 構造

`<instructions>`

[Claude への指示・依頼内容]

`</instructions>``<context>`

[背景情報・前提条件]

`</context>``<input>`

[処理してほしいデータや文章]

(次のページに続く)

(前のページからの続き)

</input>

<output_format>

[出力形式の指定]

</output_format>

タグ名は自由に決めます。上記はあくまで一例です。大切なのは「何がどこにあるか」を Claude が迷わないように区切ることです。

29.4.3 例

ビジネス寄り（顧客メールへの返信文生成）：

<instructions>

以下の顧客からのクレームメールに対する返信文を作成してください。

謝罪、原因の説明、今後の対応策の 3 つを含めてください。

丁寧なビジネス文体で、300 文字以内にまとめてください。

</instructions>

<context>

私たちは EC サイトを運営しています。

今回のトラブルの原因：物流会社のシステム障害により配送が 3 日遅延した。

対応策：次回購入時に使える 500 円クーポンを発行する予定。

</context>

<customer_email>

先注文した商品がまだ届きません。到着予定日を 3 日過ぎているのですが、

どうなっているのでしょうか。急ぎで必要だったのに困っています。

</customer_email>

開発寄り（コードレビューの自動化）：

<instructions>

以下のコードをレビューしてください。

以下の観点で問題点を指摘してください：

1. バグ・潜在的なエラー
2. パフォーマンス上の懸念
3. コーディング規約・可読性

各問題には severity (high/medium/low) を付けてください。

</instructions>

<code_language>Python 3.11</code_language>

<coding_standards>

- 関数は 1 つの責任のみ持つ

(次のページに続く)

(前のページからの続き)

```

- 変数名は snake_case
- 型ヒントを必ず付ける
</coding_standards>

<code>
def get_user_data(id):
    conn = sqlite3.connect('users.db')
    cursor = conn.cursor()
    query = f"SELECT * FROM users WHERE id = {id}"
    cursor.execute(query)
    return cursor.fetchall()
</code>

```

29.4.4 注意点

よくある失敗: タグ名が曖昧または矛盾している

<data> や <info> のような汎用的すぎるタグ名は、Claude が内容を推測しにくくなります。
<customer_complaint> や <reference_document> のように、中身を一言で表すタグ名を使いましょう。また、閉じタグ (</instructions>) を忘れないよう注意してください。

29.5 パターン 4: ペルソナ / ロール設定

29.5.1 いつ使う

- 特定の専門家視点の意見・アドバイスが欲しいとき
- 出力のトーン・文体を特定の方向に揃えたいとき
- 複数の立場からの検討が必要なとき

ロール設定は「あなたは〇〇です」と Claude に役割を与えるパターンです。ロールを与えることで、Claude は関連する知識・視点・語彙を優先的に使うようになります。

重要なのは、**ロール設定は「人格を変える」のではなく「フォーカスを当てる」もの** という理解です。「あなたは厳しい批評家です」と言っても Claude が不親切になるわけではなく、批評的な観点を前面に出した回答をしてくれます。

29.5.2 構造

あなたは [ロール] です。[ロールの具体的な背景・経験・特徴を補足]。

[タスク]

補足が重要です。「あなたはマーケターです」より「あなたは 10 年以上 BtoB SaaS のマーケティングを担当してきたプロです」のほうが、より具体的で適切な回答を得られます。

29.5.3 例

ビジネス寄り（プレゼン資料のレビュー）：

あなたは投資家向けのピッチデッキを何百件もレビューしてきたベンチャーキャピタリストです。

以下のビジネスアイデアのサマリーを読んで、
「このアイデアに投資する/しない理由」を 3 点ずつ挙げてください。
付度なしに厳しく評価してください。

[ビジネスアイデアのサマリーをここに貼る]

開発寄り（セキュリティ診断）：

あなたは Web アプリケーションのセキュリティ診断の専門家（ペネトレーションテスター）です。
OWASP のトップ 10 を熟知しており、脆弱性を見つけることが得意です。

以下の Web アプリの仕様を読んで、セキュリティ上のリスクを洗い出してください。
深刻度（Critical/High/Medium/Low）と推奨される対策もセットで記載してください。

[アプリ仕様をここに貼る]

複数のロールを使った対立的検討（メリット・デメリット分析）：

以下の新しい社内制度案について、2 つの立場からそれぞれ意見を述べてください。

制度案：週 1 回のリモートワーク全社員への導入

【推進派の立場として】

この制度の賛成理由を 3 点、具体的な効果予測とともに述べてください。

【慎重派の立場として】

この制度の懸念点を 3 点、具体的なリスクとともに述べてください。

最後に、中立的な立場として総合的な判断を述べてください。

29.5.4 注意点

よくある失敗: ロールが強すぎて事実確認が疎かになる

「あなたはその道 30 年のプロです」のような強いロール設定は、Claude が自信を持って回答する一方、不確かな情報でも断言してしまうリスクがあります。専門的な回答が必要な場合でも、「不確かな点は不確かと明示してください」という指示を添えることをお勧めします。

29.6 パターン 5: 分割と征服

29.6.1 いつ使う

- タスクが複雑で、一度に処理するには情報が多すぎる時
- 出力が長くなりすぎて品質が落ちるとき
- 各ステップの途中結果を確認・修正しながら進めたいとき

分割と征服（Divide and Conquer）は、大きなタスクを小さなサブタスクに分解して、順番に処理させるパターンです。人間が大きなプロジェクトをマイルストーンに分けるのと同じ発想です。

Claude は一度に扱う情報が増えるほど、後半の精度が落ちる傾向があります。分割することで各ステップの品質を保ちつつ、最終的に高品質な成果物を得られます。

29.6.2 構造

ステップ 1: [最初のサブタスク]

(Claude の出力を確認)

ステップ 2: 上の結果をもとに、[次のサブタスク]

(Claude の出力を確認)

ステップ 3: これまでの内容を踏まえて、[最終タスク]

あるいは、プロンプト内でステップを明示する方法もあります：

このタスクを以下のステップで処理してください。

各ステップの完了後、次のステップを開始する前に私に確認を求めてください。

ステップ 1: [サブタスク 1]

ステップ 2: [サブタスク 2]

ステップ 3: [サブタスク 3]

29.6.3 例

ビジネス寄り（市場調査レポートの作成）：

まずステップ 1 を依頼します。

新しいオンライン英会話サービスの市場調査をしています。

まず最初に、この市場の主要プレイヤー（競合他社）を 5 社挙げ、それぞれの特徴を簡潔にまとめてください。

次のステップ（価格比較・強み弱み分析）に進む前に、このリストを確認させてください。

Claude の回答を確認し、修正が必要なら調整した上でステップ 2 へ。

ありがとうございます。
では次に、先ほど挙げた 5 社の料金プランを比較し、
価格帯・提供価値の観点で整理してください。

開発寄り（新機能の設計と実装）：

ユーザー認証機能を実装したいと思います。
以下の順で進めてください。各ステップが終わったら確認を求めてください。

ステップ 1: 必要な API エンドポイントの設計 (POST /auth/register, POST /auth/login など)

ステップ 2: データベーススキーマの設計

ステップ 3: 各エンドポイントの Pseudo コード（疑似コード）の作成

ステップ 4: セキュリティ上の考慮点のレビュー

使用スタック: Node.js + Express + PostgreSQL

29.6.4 注意点

よくある失敗: 細かく分割しすぎてセッションが終わらない

タスクを 5 段階以上に分割すると、会話が長くなりすぎて途中で方向性を見失うことがあります。原則として、分割は 2~4 段階が最適です。それ以上になりそうなら、大きなステップをまとめるか、別の会話セッションで続けることを検討してください。

29.7 パターン 6: 自己検証

29.7.1 いつ使う

- 重要な成果物の品質を高めたいとき
- 最初の回答が十分かどうか自信が持てないとき
- エラーや見落としを減らしたいとき

自己検証は、Claude に自分の出力を「もう一度見直させる」パターンです。人間でも、文章を書いた後に読み返すと誤字や論理の穴に気づくことがあります。Claude も同様に、一度出力した内容を別の視点から確認させることで品質が上がります。

29.7.2 構造

方法は大きく 2 つあります。

方法 1: 一度の回答内で検証も依頼する

[タスク]

回答を生成した後、以下の観点で自分の回答を検証し、

(次のページに続く)

(前のページからの続き)

問題があれば修正したバージョンを出力してください：

- [検証観点 1]
- [検証観点 2]

方法 2: 先に回答させてから検証を依頼する

(最初のメッセージ)

[タスク]

(Claude の回答を受けて)

ありがとうございます。その回答を [観点] の視点から批判的にレビューして、改善点があれば修正版を出力してください。

29.7.3 例

ビジネス寄り（提案書のレビュー）：

以下の提案書の概要を書きました。
まず内容を読んで改善点を 5 点挙げてください。
次に、その改善点を反映した改訂版を書いてください。

【改善の観点】

- 相手（経営層）にとって何が嬉しいのかが伝わるか
- 数字・根拠が不足していないか
- 反論されそうな点を先回りして答えているか

[提案書の文章をここに貼る]

開発寄り（テストケースのレビュー）：

以下の関数のユニットテストを書いてください。

```
```python
def divide(a: float, b: float) -> float:
 return a / b
```
```

テストを書き終わったら、「テストが網羅できていないエッジケース」がないかを自分で確認してください。

見落としがあれば追加のテストケースを加えてください。

29.7.4 注意点

よくある失敗: 「何でも検証して」という曖昧な指示

検証の観点を指定しないと、表面的なチェックだけで終わることがあります。「論理的矛盾がないか」「数字の根拠が示されているか」「読者が理解しやすいか」など、具体的な観点を明示することで検証の深さが変わります。

29.8 パターン 7: マルチターン戦略

29.8.1 いつ使う

- 長い会話セッションで文脈が失われてきたとき
- 複数の話題が混在してきたとき
- 前の会話の内容を参照しながら次のタスクを進めるとき

マルチターン戦略は、複数回のやり取り（ターン）を通じた会話の流れをデザインするパターンです。単発のやり取りではなく、「会話全体をどう設計するか」という視点が必要になります。

29.8.2 構造

文脈の確認（長い会話の途中で）：

これまでの会話で合意した内容を箇条書きでまとめてください。
特に、決定した仕様と未解決の課題を分けて整理してください。

文脈の引き継ぎ（新しいセッションを始めるとき）：

前回の会話の続きです。
以下の前提条件を踏まえて作業を続けてください：
[前のセッションの要点を箇条書きでペースト]

会話の方向修正：

少し方向を変えたいと思います。
今まで話していた [テーマ A] については一旦置いておいて、
[テーマ B] にフォーカスしましょう。

29.8.3 例

ビジネス寄り（複数回の会議後の議事録作成）：

第 1 ターン：情報の収集

今週の定例会議のメモを共有します。
まずこのメモから「決定事項」「課題」「次のアクション」を
それぞれ抽出して箇条書きにしてください。

[会議メモをここに貼る]

第2ターン：加工と整形

ありがとうございます。
その整理をもとに、全員に送る議事録メールの文章を作成してください。
件名も提案してください。

開発寄り（設計レビューの段階的進行）：

第1ターン：大きな方向性の確認

新しいECサイトのバックエンド設計を相談したいです。
まず、マイクロサービスとモノリシック構成の
メリット・デメリットを比較してください。
私たちのチーム規模（5人）と想定トラフィック（月10万PV）を前提にしてください。

第2ターン：詳細設計

モノリシック構成で進めることにしました。
では次に、この規模のECサイトに必要な主要なAPIエンドポイントの
一覧を設計してください。

29.8.4 注意点

よくある失敗: 一つの会話に詰め込みすぎる

Claude は会話が長くなるほど、初期の文脈を参照しにくくなります（これは「コンテキストウィンドウ」の制限によるものです）。一つのトピックが完結したら新しい会話を始めるか、定期的に「これまでの合意事項を整理してください」と確認することをお勧めします。

29.9 パターンの組み合わせ

各パターンは単独でも使えますが、組み合わせることで真の威力を発揮します。

29.9.1 組み合わせ例 1: 競合分析レポートの作成

XML タグ構造化 + Chain of Thought + 自己検証を組み合わせています。

<instructions>

競合他社の分析レポートを作成してください。
以下の観点を順番に検討して、構造化されたレポートにまとめてください：

(次のページに続く)

(前のページからの続き)

1. 製品・サービスの特徴比較
2. 価格帯と対象顧客層
3. マーケティング戦略の傾向
4. 自社との差別化ポイント

レポートを書き終えたら、「見落としした観点がないか」「根拠が不足している主張がないか」を自分で確認して、必要に応じて加筆してください。

`</instructions>`

`<context>`

私たちは中小企業向けのプロジェクト管理 SaaS を提供しています。

主な顧客: 従業員 50~200 名の製造業・サービス業

価格帯: 月額 5,000~30,000 円 (チーム規模に応じて)

`</context>`

`<competitors>`

- Asana (米国発、大企業向けが強い)
- Backlog (日本語対応、エンジニア向け)
- Notion (ドキュメント管理寄り)

`</competitors>`

29.9.2 組み合わせ例 2: コードのバグ修正と品質改善

ロール設定 + Chain of Thought + 分割と征服を組み合わせています。

あなたはコードの品質改善を専門とするシニアエンジニアです。
以下の関数について、2つのフェーズに分けてレビューしてください。

フェーズ 1: バグと潜在的なエラーの特定

- 現在のコードで問題になりうる箇所をすべて列挙
- 各問題の深刻度 (Critical/High/Medium) を付ける
- フェーズ 1 が終わったら私に確認を求めてください

フェーズ 2: リファクタリング提案

- フェーズ 1 で合意した問題を修正した改訂版コードを提示
- 変更箇所とその理由をコメントで記載

```
```python
```

```
def send_notification(user_id, message, type):
 user = db.query(f"SELECT * FROM users WHERE id={user_id}")
 if user.email_enabled == True:
 send_email(user.email, message)
 if type == "urgent":
 send_sms(user.phone, message)
```

(次のページに続く)

(前のページからの続き)

```
log.write(f"{user_id}: {message}")
...
```

### 29.9.3 組み合わせ例 3: ビジネス企画書の作成 (Few-shot + 分割と征服)

新サービスの企画書を作成します。

以下の例と同じ構成で、各セクションを 1 つずつ作成してください。

---構成の例（既存の企画書より）---

## サービス概要

[3 行以内の要約。「誰の、どんな問題を、どう解決するか」を含める]

## 市場規模と機会

[数字を使った市場規模、成長率、参入タイミングの根拠]

## 競合との差別化

[主要競合 3 社との比較表 + 自社の独自価値]

---現在のサービス情報---

名称: WorkLog (仮)

コンセプト: 中小企業の現場作業向け日報アプリ

ターゲット: 建設業・製造業の現場監督 (スマホのみ使用)

強み: 音声入力対応、オフライン動作可能

まず「サービス概要」セクションを作成してください。

確認後、次のセクションに進みます。

## 29.10 次のステップ

パターンを理解したら、次は「どこに・どのタイミングで指示を書くか」というシステムレベルの設計に進みましょう。

- **ビジネス活用事例** (ページ 209) — 業務での具体的な Claude の活用パターンを学ぶ
- **API 活用ガイド** (ページ 357) — API を使ったシステム連携やプロンプト設計を深掘りする
- **Claude Code ワークフロー** (ページ 147) — 繰り返し作業を Claude Code で自動化するフロー設計
- **プロンプトの基本原則** (ページ 239) — このページの前編。明確さ・具体性・文脈提供の 3 原則



---

## 第30章 Claude を AI 秘書にする

メールの返信、今日やるべきことの整理、日報の作成——毎朝これだけで30分以上使っていませんか。

これらは「考える仕事」ではなく「整理・処理する仕事」です。Claude はこうした定型業務を引き受けてくれる、デジタルの右腕として機能します。チャットで質問するだけのツールではなく、毎日の秘書的タスクを任せられる相棒として使うことで、本来集中すべき仕事に時間を取り戻せます。

実践者はこんな変化を報告しています。

- 朝のタスク整理：30分 → 3分（90%削減）
- 経費精算：30分 → 5～10分（Qiita: minorun365 氏の実践）
- 未読メール：5,000通 → 優先度付き6通に絞り込み

プログラミングの知識は不要です。実際に、非エンジニアが「今まで技術的に諦めていた自動化」を Claude で実現した事例も出てきています。まず1つの業務を楽にすることから始めましょう。

### 30.1 AI 秘書に任せられる5つの業務

Claude が得意とする「秘書仕事」は大きく5つに分類できます。それぞれに専用の解説ページを用意しています。

#### 30.1.1 メール処理の自動化

受信トレイから重要なメールを抽出し、返信の下書きを作成します。「2営業日以上返信していないメール」の検出や、トーン別（フォーマル/カジュアル）の下書き生成も得意です。

→ [メール処理の自動化](#) (ページ 271)

#### 30.1.2 朝のブリーフィングとタスク整理

カレンダー・メール・Slack などの情報を統合して、「今日やるべきこと」を優先度付きでまとめます。毎朝同じ質問を投げるだけで、散在する情報が整理されます。

→ [朝のブリーフィングとタスク整理](#) (ページ 279)

#### 30.1.3 日報・週報の自動作成

作業ログやカレンダーの予定をもとに日報・週報の下書きを作ります。書く手間を減らしつつ、「振り返り」の部分は人間が行う設計が理想的です。

→ [日報・週報・報告書の自動化](#) (ページ 287)

### 30.1.4 報告書・提案書の下書き

「構造と素材を渡すだけで 80 点の下書き」が Claude の真骨頂です。バリエーション（役職別、目的別）の出し分けも簡単にできます。

→ [日報・週報・報告書の自動化](#) (ページ 287)

## 30.2 レベル別の始め方

### 30.2.1 Level 1 — 今日からできる（ブラウザだけ）

**必要なもの:** ブラウザと [claude.ai](#) のアカウント（無料）

一番簡単な始め方は、**メール文面や報告書の素材を貼り付けて下書きを作ってもらう**ことです。

**例：メールの返信下書きを作る**

以下のメールへの返信を書いてください。

【受け取ったメール】

（メール本文をここに貼り付ける）

【返信に含めたいこと】

- 来週の打ち合わせ日程を 3 つ提案する（月曜午後、水曜午前、金曜終日可）
- 添付資料は確認済みと伝える

【トーン】 フォーマル。相手は取引先の部長。

このプロンプトをコピーして、（メール本文をここに貼り付ける）の部分を実際のメールに差し替えるだけです。

### Projects に自分のプロフィールを登録する

claude.ai の「Projects」機能を使うと、自分の役職・会社・業界・よく使う文体などを一度登録しておけます。毎回「私は〇〇会社の営業部長です」と書かなくても、Claude が文脈を理解して回答します。



**注釈:**

**まず 1 つの業務だけ試してみましょう**

「全部自動化しよう」と張り切って挫折するケースが多いです。まず「毎日書いている日報」か「返信に迷うメール」のどちらか一方から始めることをおすすめします。1 週間試してみると、自分に合った使い方が見えてきます。

### 30.2.2 Level 2 — Claude Desktop を活用する（ノーコード）

**必要なもの:** Claude Desktop（無料）+ Claude Pro（月額約 3,000 円）

Claude Desktop は、パソコンにインストールするアプリ版の Claude です。**コネクタ**という機能を使うと、Gmail・Google Calendar・Google Drive などのサービスと接続できます。コードを書く必要はありません。

接続後は、こんな指示が出せるようになります。



今日の未返信メールを確認して、明日の午前中のカレンダーの空きと照合し、返信が必要なメールのリストと各返信の下書きを作ってください。

メールとカレンダーを横断した操作が、日本語の指示 1 つで完結します。



警告:

#### Gmail 連携は会社のセキュリティポリシーを確認してから

個人の Gmail アカウントであればすぐに試せますが、会社のメールアカウントを接続する前に、社内の IT・セキュリティ部門に確認してください。外部サービスとの連携に関するポリシーは会社によって異なります。

### 30.2.3 Level 3 — MCP・Claude Code で本格化する（上級者向け）

必要なもの: Claude Code + 各種 MCP サーバー

ターミナル（コマンドを入力する黒い画面）を使える方向けの、より高度な活用法です。

MCP（Model Context Protocol）という仕組みを使うと、Claude が外部サービスを直接操作できるようになります。たとえば GitHub・Slack・Todoist・Obsidianなどを統合し、「おはよう」と入力するだけで5つのサービスの情報をまとめたブリーフィングが届くような自動化が実現できます。

→ [MCP 連携・複数ツール統合](#) (ページ 299)

## 30.3 実践者の声

### 30.3.1 「朝の 30 分が 3 分になった」

アイレットの執行役員 後藤氏は、Claude Code と MCP を使って5つの情報源（メール・Slack・カレンダー・タスク管理・ノート）を統合しました。毎朝ターミナルに「おはよう」と入力するだけで、優先度付きのブリーフィングが自動生成されます。朝のタスク整理が30分から3分に短縮され、判断に集中できる時間が増えたとのことです（[iret.media](#) 記事）。

### 30.3.2 「38 プロジェクトを並行管理できるようになった」

中間管理職の morimorihogee 氏は、GitHub・GitLab・Backlog・Slack・DocBase を Claude Code で横断管理するワークフローを構築。`/project-update all` というコマンドを実行するだけで、全プロジェクトの状況が一元整理されます。コンテキストスイッチのストレスが大幅に減ったと報告しています（[TechRacho](#) 記事）。

### 30.3.3 「5,000 通の未読メールが 6 通になった」

研究者の Chris Blattman 氏は、9つのスキルを統合した AI エグゼクティブアシスタントを構築し、5,000 通の未読メールを重要な 6 通まで絞り込みました。「人間関係が重要なメールは人間が書く」という線引きを設けたことで、AI への過度な依存を防ぎながら週 1〜2 時間の時短を実現しています（[claudeblattman.com](#) 記事）。

## 30.4 AI 秘書の限界と使い分け

Claude に何でも任せればよい、というわけではありません。以下の使い分けが効果的です。

### Claude に任せるべきこと

- 定型フォーマットへの情報整理（日報、報告書、議事録）
- 複数の情報源からの要約・抽出
- 文章の下書き・バリエーション生成
- スケジュールや期日の把握・整理

### 人間が行うべきこと

- 意思決定と最終判断
- 人間関係が重要なメール・コミュニケーション
- 振り返り・省察（「なぜうまくいったか/いかなかったか」）
- 戦略的な優先順位付け

#### つまづきポイント：Claude が「別人のような文体」でメールを書く

Claude にメールを書かせると、自分らしくない硬い文体になることがあります。解決策は2つ。  
①プロンプトに「私はカジュアルな文体で書きます。語尾に『～ですね』をよく使います」と書き添える。②過去の自分のメールをいくつかコピーして「この文体を参考に書いてください」と一緒に渡す。

## 30.5 もっと詳しく

目的に合わせて、以下のページで実践ワークフローを学んでください。

### 30.5.1 メール処理の自動化 (ページ 271)

返信下書き・優先度分類・未返信の検出まで。Level 1（コピペ）から Level 3（Gmail 自動連携）まで段階的に解説します。

### 30.5.2 朝のブリーフィングとタスク整理 (ページ 279)

カレンダー・メール・Slack を統合した「今日のやること一覧」を自動生成する方法。Claude Desktop のコネクタを使ったノーコード実装を中心に紹介。

### 30.5.3 日報・週報・報告書の自動化 (ページ 287)

作業ログからの日報自動生成、報告書・提案書の下書き作成テンプレート。「日報は気合いじゃなく仕組みで続ける」設計思想を解説。

### 30.5.4 MCP 連携・複数ツール統合（上級）（ページ 299）

複数サービスを統合した朝の自動ブリーフィング、タスク管理ツール連携、Claude Code を使った本格的な AI 秘書の構築方法。

## 30.6 よくある質問

### 30.6.1 無料版でも使えますか？

Level 1 の活用（コピペでのメール下書き・報告書作成）は Claude 無料版で十分に使えます。毎日ヘビーに使うなら Claude Pro（月額約 3,000 円）への移行をおすすめします。Level 2 以降（Claude Desktop のコネクタ機能）は Pro プランが必要です。

### 30.6.2 会社の情報を Claude に送っても大丈夫ですか？

Anthropic の方針では、claude.ai での会話はデフォルトでモデルの学習に使用されません。ただし、社内のセキュリティポリシーとの整合は必ず確認してください。顧客名・金額・個人名など機密性の高い情報は、仮名や「株式会社 A」のようにマスキングしてから使用することをおすすめします。

### 30.6.3 どこから始めればいいかわかりません

まずは「今日書いた（または書く予定の）メール 1 通」を Claude に下書きさせてみてください。Level 1 のプロンプトをそのままコピーして、自分のメール情報を貼り付けるだけです。所要時間は 5 分以内。それだけで AI 秘書の価値が実感できます。

## 30.7 関連記事

- [プロンプトの基本テクニック](#)（ページ 239） — Claude により正確に伝わる指示の書き方



---

## 第31章 メール処理を Claude で効率化する

1日に何十通ものメールを捌いている方にとって、メールは「作業の合間にやること」ではなく「作業そのもの」になっている場合が少なくありません。Claudeを使うと、返信の下書き・優先度の分類・未返信の検出まで、メール業務のかなりの部分を効率化できます。

この記事では、今日から始められる **Level 1 (コピペ)** から、Gmail と連携する **Level 2 (コネクタ)**、本格的な自動化の **Level 3 (MCP)** まで、段階的に解説します。

### 31.1 なぜメール処理はこんなに時間がかかるのか

メール業務にかかる時間を分解すると、大きく3つの負担があります。

- 1. 分類の負担** — 受信トレイを開いて「これは今すぐ返すべきか、あとでいいか、不要か」を判断する作業。1通30秒でも50通で25分かかります。
- 2. 判断の負担** — 「この依頼、どう返せばいいか」を考える作業。特に断るとき、期待値を調整するとき、複数の選択肢を提示するときは、文章を書く前の「考える時間」が長くなりがちです。
- 3. 文面作成の負担** — 考えた内容を、適切なトーンで、読みやすい文章に整える作業。「伝わるか」「失礼ではないか」を確認しながら書くため、思っているより時間がかかります。

Claude は **判断と文面作成** を大幅に補助できます。「どう返すか」の方向性を決めてプロンプトに渡せば、Claude がドラフトを作ります。あとは確認して送るだけです。



警告:

#### AI が書いたメールはそのまま送らない

Claude が作った下書きはあくまでドラフトです。送信前に必ず自分で読み、事実確認・トーンの調整・署名の確認を行ってください。特に数字・日程・固有名詞は必ずチェックしてください。

### 31.2 Level 1: コピペで始めるメール下書き

**必要なもの:** [claude.ai](#) のアカウントだけ。Claude Desktop もインストール不要です。

#### 31.2.1 新規メールの下書きを作る

用件を箇条書きで渡して、Claude にメールの形に整えてもらいます。

**フォーマルな新規メール (社外・初めての相手) :**

以下の内容で、丁寧なビジネスメールを作成してください。

【相手】株式会社〇〇 営業部 田中様（初めてご連絡する相手）

【用件】弊社サービスのデモ依頼

【伝えたいこと】

- 先日のウェビナーでサービスを知り、興味を持った
- 30 分程度のデモをお願いしたい
- 来週以降で都合の良い日時を教えてください

【自分の情報】

- 名前：山田 太郎
- 会社名：株式会社〇〇
- 部署：経営企画部

**カジュアルなメール（社内・よく知っている相手）：**

以下の内容で、社内向けのカジュアルなメールを作成してください。  
堅苦しくなりすぎず、読みやすい文体にしてください。

【相手】同じ部署の鈴木さん

【用件】来週の打ち合わせ日程の調整

【伝えたいこと】

- 来週火曜の 14 時か水曜の 11 時で都合を聞きたい
- 場所は第 3 会議室を仮押さえしている
- 打ち合わせの内容は来期のプロモーション計画

### 31.2.2 受信メールへの返信下書きを作る

元のメールをそのまま貼り付けて、「どう返したいか」を一言添えるだけです。

**依頼を丁寧に断る場合：**

以下のメールに対して、丁重にお断りする返信を書いてください。

断る理由：現在のリソースが不足しており、品質を担保できないため。

別の機会への含みを持たせてください。

---（元のメールをここに貼り付ける）---

**日程を提案する場合：**

以下のメールに対して、打ち合わせ日程を提案する返信を書いてください。

提案できる日時：

- 来週月曜 14:00～16:00
- 来週水曜 10:00～12:00
- 来週金曜 15:00～17:00

オンライン（Google Meet）を希望します。

(次のページに続く)

(前のページからの続き)

--- (元のメールをここに貼り付ける) ---

### 31.2.3 英語メールの返信

英語のメールが来たとき、「何が書いてあるか」の理解と「どう返すか」の文面作成を同時に頼めます。

以下の英語メールを日本語で要約し、その後に英語で返信を作成してください。

返信の内容：

- ミーティングの提案を承諾する
- 来週水曜 10:00 JST を提案する
- Zoom のリンクを自分から送ると伝える

--- (英語のメールをここに貼り付ける) ---

### 31.2.4 Projects に自分のメールスタイルを登録する

[claude.ai の Projects 機能](#) (ページ 89) を使うと、毎回プロンプトで説明しなくてもよくなります。

**Projects に登録しておく便利な情報:**

#### 【メール対応の基本情報】

- 私の名前：山田 太郎
- 会社名・部署：株式会社〇〇 経営企画部
- メールのトーン：丁寧だが堅苦しすぎない。「させていただく」の多用は避ける
- 署名：  
山田 太郎 / 株式会社〇〇 経営企画部  
Email: yamada@example.com / TEL: 03-XXXX-XXXX

#### 【よく使うフレーズ】

- 締め：「ご確認のほど、よろしくお願いいたします。」
- 返信遅れ時：「ご連絡が遅くなり失礼いたしました。」

これを Project の指示（システムプロンプト）に登録しておくと、その Project 内での会話では自動的にこの情報が使われます。



**注釈:**

#### Projects の作り方

claude.ai の左サイドバーから「Projects」→「New project」で作成できます。「Project instructions」に上記のような情報を貼り付けておくと、この Project 内でのやりとり毎に毎回適用されます。詳しくは [Projects の使い方](#) (ページ 89) を参照してください。

## 31.3 Level 2: Claude Desktop コネクタで Gmail 接続

**必要なもの:** Claude Desktop（無料インストール） + Google アカウント

Claude Desktop にコネクタを設定すると、Claude が直接 Gmail を読んだり書いたりできるようになります。「メールを見せて」と頼むだけで受信トレイを確認し、「この件について返信案を作って」と頼むだけで下書きを作成できます。



**警告:**

### セキュリティの確認を先に

コネクタを設定する前に、会社のセキュリティポリシーで Google アカウントを外部ツールと連携することが許可されているか確認してください。業務用アカウントで連携する場合は IT 部門への確認を推奨します。個人の Gmail アカウントで試す場合でも、機密情報が含まれるメールをそのまま Claude に渡すことのリスクを理解した上で使用してください。

### 31.3.1 Gmail コネクタの設定手順

1. Claude Desktop を開き、左下の設定アイコンをクリック
2. 「Integrations」または「Connectors」セクションを探す
3. 「Gmail」を選択して「Connect」をクリック
4. Google アカウントでの認証画面が表示されるので、連携を許可する

設定後は、チャット画面で自然な言葉でメールを操作できます。

**コネクタを使った操作例:**

今日の未読メールを確認して、返信が必要なものを優先度順に教えて

田中さんからのメールを確認して、返信案を作って。  
丁寧なトーンで、来週水曜の打ち合わせを承諾する内容で。

### 31.3.2 メール確認 → カレンダー照合 → 返信案の一気通貫フロー

Google Calendar コネクタも設定すると、「空き時間を確認しながら日程調整メールの返信を作る」が一気通貫でできます。

**note:** Claude Desktop のコネクタで Gmail + Calendar を自動化 — ノーコードで実現できるこの連携について、もりたりく氏が具体的な操作手順を解説しています。

**実際の操作例:**

田中さんから来週の打ち合わせ日程調整メールが来ています。  
来週の私のカレンダーを確認して、空いている時間帯をもとに  
返信案を作ってください。

Claude はメールを読み、カレンダーを確認し、空き時間を踏まえた返信案を作成します。最後に「この返信でいいですか？」と確認を求めてくるので、問題なければ送信します。



### 31.3.3 未返信メールの検出

2 営業日以上返信していないメールを見落としていることがありますか？コネクタを使えばこれを検出できます。

受信トレイを確認して、2 営業日以上返信していないメールを教えてください。  
相手と件名、受信日、どんな内容だったかを一覧にしてください。

## 31.4 Level 3: MCP で本格的なメール管理

**必要なもの:** Claude Code またはエンジニアのサポート

MCP (Model Context Protocol) を使うと、より高度なメール管理が可能になります。ここはエンジニア向けの内容です。非エンジニアの方は [Level 2](#) (ページ 274) の活用から始めることをおすすめします。

### 31.4.1 Inbox Triage (受信トレイの自動分類)

MCP を使ったメール管理の代表的な活用が「Inbox Triage (トリアージ)」です。受信トレイのメールを自動的に以下の 4 カテゴリに分類します。

カテゴリ	内容
Action Required	返信・判断・タスクが必要なメール
Waiting	相手の返信待ちで経過観察が必要なメール
FYI	読むだけでいい情報共有メール
Promotional	削除・アーカイブ可能なプロモーション

Chris Blattman 氏の事例では、このアプローチを MCP を組み合わせた AI エグゼクティブアシスタントの構築に活用し、**5,000 通の未読メールを 6 通まで削減**しました ([Building an AI Executive Assistant - Claude Blattman](#))。

### 31.4.2 ポリシーファイルで AI の行動を制御する

自律的なメール処理の設計で重要なのが「どこまで AI に任せるか」の設計です。Blattman 氏の設計では、以下のような段階的自律性を採用しています。

- **提案フェーズ:** AI が分類・優先度付け・返信案を作成。人間が確認・承認
- **承認フェーズ:** 定型的な返信（確認のお礼、日程承諾など）は AI が自動送信。重要なものは人間が判断
- **自動化フェーズ:** ルーティンメールの分類・アーカイブを自動化

この設計は POLICY.md や CLAUDE.md などのポリシーファイルに記述し、AI の行動範囲を明示的に定義します。

# メール対応ポリシー

(次のページに続く)

(前のページからの続き)

### ## 自動で実行してよいこと

- 受信メールの分類 (Action Required / FYI / Promotional)
- ニュースレターとプロモーションメールのアーカイブ
- 返信下書きの作成 (送信前に必ず確認を求める)

### ## 必ず人間の確認を取ること

- メールの実際の送信
- 機密情報が含まれるメールの処理
- 初めてやりとりする相手へのメール

## 31.5 「人間が書くべきメール」の線引き

Claude はメール作業の多くを助けてくれますが、**すべてのメールを AI に任せてよいわけではありません。**

### 31.5.1 AI に下書きを任せてよいメール

- 日程調整・確認
- 情報共有・お知らせ
- 定型的な依頼・受諾
- 英語メールの返信

### 31.5.2 人間が自分で書くべきメール

**謝罪メール** — 何かミスや問題が起きた際の謝罪は、誠意が伝わるのが最優先です。AI が書いた文章は形式的に見えることがあり、かえって相手の感情を逆なでする可能性があります。

**交渉メール** — 価格交渉、契約条件の調整、重要な取引の交渉は、ニュアンスや駆け引きが重要です。AI は「合理的な」文章を書きますが、交渉に必要な「間合い」や「余白」を読むのは難しいです。

**人間関係が重要なメール** — 長年の取引先への感謝、お見舞い、お悔やみなど、相手との関係性が重要なメールは自分の言葉で書いてください。

**会社を代表する重要な表明** — 経営判断の発表、公式見解の表明、法的な意味を持つメールは、専門家の確認を経た上で人間が書くべきです。

#### 「AI が書いたメールはバレるか？」という疑問

ビジネスパーソンの中で「AI が書いた文章は分かる」という感覚が広まっています。完全に AI に任せた文章は、なめらかすぎる・均質すぎる・「自分らしさ」がないという特徴が出やすいです。Claude はあくまで **下書きを作るアシスタント** として使い、自分の言葉で加筆・修正することをおすすめします。

## 31.6 コピペで使えるプロンプト集

### 31.6.1 プロンプト 1: 新規メールの下書き（汎用）

以下の情報をもとに、ビジネスメールを作成してください。

【相手の情報】

- 宛先：【会社名・部署・名前】
- 関係性：【初めて / 既知・社外 / 社内】

【メールの目的】

【例：デモのアポを取りたい / 資料を送付したい / 質問したい】

【伝えたい内容（箇条書きで）】

- 
- 
- 

【自分の情報】

- 名前：
- 会社・部署：

【トーン】

【フォーマル / 通常のビジネス / カジュアル】

### 31.6.2 プロンプト 2: 受信メールへの返信

以下のメールへの返信を作成してください。

【返信の方向性】

【例：承諾して日程を提案する / 丁重に断る / 追加情報を要求する】

【補足事項（ある場合）】

【例：提案できる日時、断る理由など】

---（元のメールをここに貼り付ける）---

### 31.6.3 プロンプト 3: 英語メールの返信

以下の英語メールを日本語で要約した後、英語で返信を作成してください。

【返信で伝えたいこと】

- 
- 

(次のページに続く)

(前のページからの続き)

【トーン】 Professional / Friendly-professional

--- (英語メールをここに貼り付ける) ---

### 31.6.4 プロンプト 4: メールの優先度分類 (コピペで使う場合)

以下のメール一覧を、以下のカテゴリに分類してください。

カテゴリ：

- [赤] 今日中に対応が必要
- [黄] 今週中に対応が必要
- [緑] 確認だけで OK (返信不要)
- [白] 不要 (削除・アーカイブ可)

--- (メールのリストや件名・本文をここに貼り付ける) ---



警告:

#### 機密情報の取り扱いに注意

メール本文を Claude に貼り付ける際は、以下の点に注意してください：

- **社外秘・機密情報が含まれるメール**は貼り付けない。必要な部分だけを要約して伝える
- **個人情報** (顧客の連絡先、社員情報など) は必要最小限に留める
- **会社のセキュリティポリシー**で AI サービスへの情報入力制限されている場合は従う
- Claude への入力内容は、**プライバシーポリシー**に基づいて取り扱われます

## 31.7 次のステップ

メール処理の効率化ができれば、他の業務にも AI 秘書を広げてみましょう。

- **朝のブリーフィングとタスク整理** (ページ 279) — 今日やるべきことを 3 分で把握する方法
- **日報・週報・報告書の自動化** (ページ 287) — 作業ログから報告書を自動生成する方法
- **AI 秘書の全体像** (ページ 265) — Level 1〜3 の全体像と実践者の事例

---

## 第32章 朝のタスク整理を自動化する

朝のルーティンを振り返ってみてください。

メールを開いて未読を確認する。カレンダーを開いて今日の予定を把握する。Slack を開いてメンションを確認する。タスク管理ツールを開いて昨日からの積み残しを確認する。——これを毎朝繰り返すだけで、気づけば 30 分以上が経っていることはありませんか。

情報が散在しているのが問題です。メール・カレンダー・チャットツール・タスクツールをそれぞれ別に確認し、頭の中で統合して「今日やること」を決める——この「頭の中での統合作業」こそ、最もエネルギーを消耗するステップです。

Claude はこの統合作業を代行できます。バラバラの情報を一度に渡せば、優先度付きの「今日のアクションリスト」を数分で作ってくれます。実践者の一例では、この作業が **30 分から 3 分に短縮**されています。

本記事では、今日すぐ始められるコピペ活用から、ターミナル 1 コマンドで全自動化する上級テクニックまで、3 段階で解説します。

### 32.1 Level 1: claude.ai でコピペ・ブリーフィング

**必要なもの:** ブラウザと [claude.ai](#) のアカウント（無料）

最初のステップはシンプルです。今日の情報をコピーして Claude に貼り付け、「今日の優先事項を教えて」と頼むだけです。

#### 32.1.1 カレンダーをコピペして優先事項を整理する

Google カレンダーや Outlook の今日の予定をテキストでコピーし、以下のプロンプトに貼り付けます。

【今日の予定】

（カレンダーの内容をここに貼り付ける）

上記の予定をもとに、以下を整理してください：

1. 準備が必要な会議と、各会議で確認・決定すべきこと
2. 予定と予定の間にできる作業タスク（30 分以内でできるもの）
3. 今日中に絶対終わらせるべき優先事項（上位 3 つ）

私の役割：（例：マーケティング部のマネージャー、3 名のチームを担当）

これだけで、「14 時の定例前に資料の最終確認を 15 分確保する」「午前中の空き時間で先週の企画書レビューを終わらせる」といった具体的な行動計画が出てきます。

### 32.1.2 未読メール一覧をコピーして優先度を付ける

受信トレイの未読メール一覧（差出人・件名・日時）をコピーし、以下のプロンプトに貼り付けます。

【今日の未読メール一覧】

（差出人、件名、受信日時をここに貼り付ける）

以下の基準で優先度を付けて整理してください：

- 高：今日中に返信が必要なもの、承認・判断が必要なもの
- 中：今週中に対応すればよいもの
- 低：参考情報・FYI・Newsletter 等

各メールへの対応方針（1 行）も一緒に書いてください。

「差出人と件名だけ」でも十分です。メール本文を貼り付ける必要はありません。

#### つまづきポイント：コピーしたテキストが崩れる

カレンダーやメールクライアントから貼り付けると、書式や記号が崩れることがあります。その場合は「差出人：〇〇、件名：〇〇、日時：〇〇」の形式で手入力するか、箇条書きでざっくり書き直しても構いません。Claude は完璧なフォーマットがなくても内容を読み取れます。

### 32.1.3 Projects に役割情報を登録しておく

毎回「私はマーケティング部のマネージャーで……」と説明するのは手間です。claude.ai の **Projects** 機能を使えば、自分の役割・担当業務・定例会議のスケジュールを一度登録しておけます。

Projects の「Project instructions」（プロジェクト指示）に以下のような情報を書いておきましょう。

【私の基本情報】

- 役割：マーケティング部 シニアマネージャー
- チーム規模：5 名
- 担当：デジタルマーケティング全般、外部パートナーとのコミュニケーション

【定例会議】

- 月曜 10 時：部門全体朝会（30 分）
- 水曜 15 時：経営会議（1 時間）
- 毎日 9 時：チームスタンドアップ（15 分）

【よく使うツール】

- Slack（社内コミュニケーション）
- Asana（タスク管理）
- Google Workspace（メール・カレンダー）

この情報があると、カレンダーをコピーするだけで「水曜の経営会議に向けて、月・火で資料を仕上げる必要がある」といった文脈を読んだ提案をしてくれるようになります。



注釈:

#### 毎朝同じプロンプトを使い回す

一度「自分に合ったブリーフィングプロンプト」ができれば、テキストファイルやメモアプリに保存しておきましょう。毎朝そのプロンプトをコピーして情報だけ差し替えれば、5分以内に今日のアクションリストが揃います。

## 32.2 Level 2: Claude Desktop のコネクタで自動収集

**必要なもの:** Claude Desktop（無料）+ Claude Pro（月額約 3,000 円）

Level 1 ではカレンダーやメールを手動でコピーしていましたが、Claude Desktop のコネクタ機能を使うと、Claude が直接 Gmail・Google Calendar にアクセスできるようになります。コードは一切不要です。

### 32.2.1 Gmail と Google Calendar のコネクタを設定する

Claude Desktop を起動し、設定画面から「Integrations」を開きます。Gmail と Google Calendar のコネクタを追加し、Google アカウントで認証してください。



警告:

#### 会社のメールアカウントを使う場合は注意

個人の Google アカウントであればすぐに試せますが、会社のメールやカレンダーを接続する前に、社内の IT・セキュリティ部門に確認してください。外部サービスへの接続に関するポリシーは会社によって異なります。

### 32.2.2 一言でブリーフィングを生成する

コネクタの設定が完了したら、以下のように指示するだけです。

今日のブリーフィングを作ってください。

確認してほしいこと：

1. 今日のカレンダー予定（準備が必要なものに★を付ける）
2. 未読メールの優先度付きリスト（高・中・低）
3. 今日中に対応が必要なアクションをまとめたリスト

Claude が Gmail と Google Calendar を確認し、まとめて返答してくれます。毎朝このプロンプトをそのまま使い回せます。

### 32.2.3 空き時間にタスクを割り当てる

コネクタがあると、カレンダーの空き時間を把握した上でタスクを割り当てる提案もできます。

今日のカレンダーを確認して、空き時間にこのタスクリストを割り当ててください：

- 企画書レビュー（60 分）
- 請求書の確認と承認（30 分）

(次のページに続く)

(前のページからの続き)

- 来週の週次報告の準備 (45 分)

優先度の高いものを午前中に入れてください。

「14 時～15 時の空きに企画書レビュー、16 時以降に来週の報告準備を入れると良さそうです」といった、実際のスケジュールに基づいた具体的な提案が返ってきます。

### つまづきポイント：コネクタがうまく動かない

コネクタ設定直後は Google の認証に数分かかることがあります。「メールにアクセスできません」というエラーが出た場合は、Claude Desktop を再起動してから試してみてください。それでも解決しない場合は、Settings → Integrations から一度コネクタを削除して再設定すると解決することが多いです。

## 32.3 Level 3: 「おはよう」で全自動ブリーフィング

**必要なもの:** Claude Code + 各種 MCP サーバー + ターミナルの基本知識

ここからはエンジニアリングが必要なレベルの話です。「ターミナル (コマンドを入力する黒い画面) を操作したことがある」ことが前提になります。非エンジニアの方には難易度が高いですが、実現できることの規模が一段と大きくなります。

### 32.3.1 「おはよう」で 5 サービスを統合する (iret の事例)

アイレットの執行役員・後藤和貴氏は、Claude Code と MCP を組み合わせて、ターミナルに「おはよう」と入力するだけで 5 つのサービスの情報をまとめたブリーフィングが届く仕組みを構築しました (iret.media の記事 で詳しく紹介されています)。

統合されているサービスは Gmail ・ Google Calendar ・ Notion ・ Backlog ・ Slack です。毎朝 Claude が各サービスにアクセスして以下の情報を収集し、まとめて提示してくれます。

- 未読メール (2 営業日以上返信していないものを特別に検出)
- 今日のカレンダー予定と準備事項
- Slack のメンション・重要チャンネルの動き
- タスク管理ツールの積み残しと新着タスク
- ノートアプリの昨日の記録サマリー

そして重要なのが、これらを 3 段階の**優先度** (赤・黄・緑) で自動分類して表示する点です。「赤は今日中に対応必須」「黄は今週中」「緑は確認だけ」という基準が設定されているため、ブリーフィングを受け取ったら赤から順に対応するだけです。

結果として、朝のタスク整理にかかる時間が **30 分から約 3 分**に短縮されたと報告されています。

この仕組みを実現するためには、以下が必要です。

- **Claude Code** のインストール (ターミナルから使う Claude)
- **MCP サーバー** の設定 (各サービスへの接続役)



- カスタム **CLAUDE.md** (朝のブリーフィングの指示を記述するファイル)



警告:

#### セットアップには数時間～数日かかります

「おはよう」で全自動という完成形は魅力的ですが、そこに至るまでのセットアップは決して簡単ではありません。MCP サーバーの設定、各サービスの API 認証、動作確認と微調整——これらには最低でも半日、慣れていない方であれば数日かかります。最初から完全自動化を目指さず、まず Level 1 や Level 2 で効果を実感してから挑戦することをおすすめします。

### 32.3.2 GTD + Zettelkasten で /morning コマンド (クラスメソッドの事例)

クラスメソッドのきんじょー氏は、GTD (Getting Things Done) と Zettelkasten (カード型ノート術) という 2 つの知識管理手法を Claude Code で統合したワークフローを公開しています ([dev.classmethod.jp](https://dev.classmethod.jp) の記事)。

このワークフローでは、独自のコマンドを定義しています。

- /morning — 朝のプランニング。「今日何をすべきか」を整理
- /inbox — 未処理のメモや思考を整理する「インボックス処理」
- /review — 週次レビュー。今週の振り返りと来週の準備

特に注目したいのは、この設計が単なる情報収集にとどまらず、「考える」から「判断する」へのシフトを目指している点です。朝の時間を「何をやるべきか悩む時間」から「Claude が整理した選択肢に対して Yes/No を判断する時間」に変えることで、認知負荷を大幅に下げています。

GTD や Zettelkasten に詳しくない方でも、「朝に /morning を実行して Claude に今日の計画を提案してもらい、自分で承認・修正する」という流れは参考になります。

### 32.3.3 MCP サーバーの構成例

Level 3 の自動ブリーフィングを実現するために必要な MCP サーバーの代表例を紹介します。自分の使っているサービスに合わせて選んでください。

用途	MCP サーバー	備考
Gmail・Calendar 連携	Google Workspace MCP	Anthropic 公式の連携先
Slack 連携	Slack MCP	チャンネル読み取り・メンション確認
GitHub 連携	GitHub MCP	Anthropic 公式提供
Notion 連携	Notion MCP	ノート・データベース読み書き
タスク管理全般	Todoist MCP / Linear MCP	使用ツールに応じて選択

MCP の詳細なセットアップ方法は、[MCP 連携・複数ツール統合 \(上級\)](#) (ページ 299) で解説しています。

## 32.4 週次レビューの自動化

毎朝のブリーフィングと同じ考え方で、**週次レビュー**も自動化できます。週末や金曜の終業前に以下のプロンプトを実行してみてください。

今週のレビューを手伝ってください。

【今週のカレンダー予定】

(今週参加した会議・予定の一覧を貼り付ける)

【完了したタスク】

(完了済みタスクの一覧を貼り付ける)

【積み残し・持ち越し】

(未完了タスクを貼り付ける)

上記をもとに：

1. 今週の主な成果と進捗 (3 行)
  2. 来週に繰り越す重要タスク (優先度順)
  3. 来週の最初に確認すべきこと
- をまとめてください。

週報の素材になるだけでなく、月曜の朝を「先週の続きを思い出すところから始める」状態から脱却できます。



注釈:

**週次レビューは「振り返り」まで自動化しない**

Claude に「今週の反省点を教えて」と聞くことはできますが、「なぜうまくいったか・いかなかったか」の省察（内省）は、自分で行う価値があります。Claude は事実の整理を担い、そこから何を学ぶかは人間が考える——この分担が、週次レビューを意味のある習慣にします。

## 32.5 実践者のワークフロー比較

### 32.5.1 ired（後藤氏）のアプローチ

5 サービスを MCP で統合し、1 コマンドで全情報を収集。優先度の 3 段階自動分類が特徴。**最大の強み**は「判断不要」で朝を始められる点。収集 → 分類 → 提示まで完全自動化されているため、受け取ったら赤から対応するだけ。**課題**は初期セットアップの複雑さと、各サービスの API 認証の維持管理。

### 32.5.2 クラスメソッド（きんじょー氏）のアプローチ

GTD + Zettelkasten という知識管理フレームワークと Claude Code を統合。コマンド体系（/morning, /inbox, /review）が整備されており、ルーティンとして定着しやすい。**最大の強み**はタスク管理だけでなくナレッジ管理も統合している点。**課題**は GTD の考え方を理解していないと効果が出にくいこと。

### 32.5.3 Chris Blattman 氏のアプローチ

「段階的な自律性」という設計思想が特徴的です。最初は「提案するだけ」から始め、信頼できると判断したものだけを「自動実行」にステップアップします。メール対応では「人間関係が重要なメールは人間が書く」という明確な線引きを設けています ([claudeblattman.com](https://claudeblattman.com) の記事)。

この3者に共通しているのは、「**Claude が整理し、人間が判断する**」という役割分担を明確にしていることです。完全に丸投げするのではなく、Claude が下準備をして、人間が最終的な優先順位を決める——この設計が長期的に機能するワークフローの鍵です。

## 32.6 今日から始めるための最初の一步

どのレベルから始めるかに迷ったら、以下を参考にしてください。

**「今日から始める」なら Level 1 から。**今朝のカレンダーをコピーして「今日の優先事項を教えて」と聞くだけです。所要時間は5分以内。まずこれを試してみてください。

**「もう少し自動化したい」なら Level 2 へ。**Claude Desktop のコネクタ設定は30分程度で完了します。一度設定すれば毎朝1文で指示するだけになります。

**「本格的に自動化したい」なら Level 3 へ。**ただし、まず Level 1 と Level 2 で自分に合ったブリーフィングの形を見つけてからにすることを強くおすすめします。「何が欲しいか」が明確でないと、Level 3 のセットアップ中に迷子になります。

#### つまづきポイント：ブリーフィングの内容が毎日同じになってしまう

「今日の優先事項」を Claude に聞いても、毎日似たような答えが返ってきて役に立たないと感じる場合があります。原因はほとんどの場合、プロンプトに「今日の文脈」が入っていないことです。「昨日終わらなかったこと」「今週の最重要目標」「今月の締め切り」など、動的な情報を毎回渡すようにしましょう。Projects にテンプレートを置いておき、そこに今日の情報を書き足す形が運用しやすいです。

## 32.7 次のステップ

- **メール処理の自動化** (ページ 271) — 返信下書きの作成、優先度分類、未返信メールの検出
- **日報・週報・報告書の自動化** (ページ 287) — 作業ログから日報を自動生成する仕組み
- **MCP 連携・複数ツール統合 (上級)** (ページ 299) — Level 3 の詳細なセットアップ方法
- **Claude を AI 秘書にする (概要)** (ページ 265) — AI 秘書活用の全体像とレベル別の始め方



---

## 第33章 日報・週報・報告書を Claude で作る

「今日も日報を書かないといけない。でも何を書けばいいか毎回迷う」

この感覚、心当たりがあれば、Claude が解決してくれます。日報・週報・報告書の「書く」部分は Claude に任せて、あなたは「振り返り」と「判断」に集中する——それが、この記事で紹介する使い方です。

### 33.1 日報・週報が続かない本当の理由

日報が続かないのは、意志が弱いからではありません。「気合い」の問題ではなく、「仕組み」の問題です。

OPTEMO のエンジニア田代氏は、2020 年から毎日日報をつけ続けた経験をもとに「日報は気合いじゃなく仕組みで続ける」と語っています。毎日続けられる人は、書くための仕組みを持っているだけです。

続かない理由は大きく 2 つあります。

1. 白紙から書き始めるコスト — 何を書けばいいか毎回考えなければならない
2. 整形の手間 — 箇条書きメモを「読みやすい日報」に整える時間が惜しい

Claude はこの 2 つをまとめて解決します。あなたが今日やったことを箇条書きで貼り付けるだけで、日報の形式に整えてくれます。

### 33.2 Level 1：コピペで日報を作る

#### 33.2.1 基本の使い方

今日やったことをメモしておいて、それを Claude に渡すだけです。

手順：

1. 工作中、やったことを箇条書きでメモしておく（Slack、メモアプリ、なんでも可）
2. 終業前に claude.ai を開く
3. 下のテンプレートをコピーして、メモを貼り付ける

今日の作業ログをもとに日報を作成してください。

【今日やったこと（メモ）】  
（ここに箇条書きで貼り付ける）

【フォーマット】  
- 今日の主な業務（3～5 項目）

（次のページに続く）

(前のページからの続き)

- 進捗・成果
- 課題・懸念事項
- 所感
- 明日の予定

【条件】

- 社内向け。読者は直属の上長
- 全体で 300～400 字程度
- 箇条書きと文章を組み合わせる



注釈:

メモはどんな形式でも OK

「A 社の資料修正。B さんに確認依頼。定例ミーティング。メール 10 通」のような断片的なメモで十分です。Claude が前後の文脈を補いながら、読みやすい日報に整形します。

### 33.2.2 日報テンプレート（コピペで使える）

以下のテンプレートをそのままコピーして、() の部分を自分の情報に差し替えてください。

以下のメモをもとに、日報を作成してください。

【メモ】

(今日やったことを箇条書きで)

【私の役職・業務】

(例：営業部 / 法人向け新規開拓担当)

【読者】

(例：部長)

【求めるフォーマット】

■ 本日の業務

(箇条書きで 3～5 項目)

■ 成果・進捗

(数字や具体的な変化があれば記載)

■ 課題・懸念

(なければ「特になし」で可)

■ 所感・気づき

(2～3 文)

(次のページに続く)

(前のページからの続き)

■ 明日の予定  
(箇条書きで2〜3項目)

【文字数】300〜400字

### 33.2.3 週報テンプレート（コピペで使える）

以下の素材をもとに、週報を作成してください。

【今週の作業メモ（月曜〜金曜）】  
(各日のメモを箇条書きで)

【私の役職・業務】  
(例：マーケティング部 / SNS 担当)

【読者】  
(例：チームリーダー・部長)

【求めるフォーマット】

■ 今週の成果・完了タスク

■ 進行中のタスクと進捗率

■ 課題・来週に持ち越す問題

■ 来週の計画（主要タスク）

■ 所感・学び

【文字数】500〜700字

**つまづきポイント：**Claude が「よく頑張りました」のような感想を書いてくる

プロンプトに「感想・励ましの言葉は不要です。事実ベースで書いてください」と一言追加することで解決します。ビジネス文書らしいドライなトーンになります。

## 33.3 Level 2：カレンダー・チャットから自動生成する

### 33.3.1 カレンダーの予定を日報に変換する

Google カレンダーの予定一覧をコピーして、そのまま Claude に渡す方法です。

手順：

1. Google カレンダーの「日表示」または「週表示」を開く
2. その日（週）の予定をすべてテキストでコピーする

### 3. 以下のプロンプトに貼り付けて送る

以下は私の今日のカレンダー予定です。  
これをもとに日報を作成してください。  
実際に参加した会議・作業の内容を推測しながら、  
ビジネス文書として自然な日報に整えてください。

【カレンダー予定】

(カレンダーの予定をここに貼り付ける)

【追記事項】

(カレンダーに載っていない作業があれば箇条書きで)

■ フォーマット：

- 本日の主な業務 (3〜5 項目)
- 成果・進捗
- 明日の予定
- 所感



注釈:

予定名だけでも OK

「10:00 A 社定例」「14:00 採用面接×2 名」のような予定名だけでも、Claude はそこから合理的な内容を推測して文章化します。「会議の内容は自分で修正する」という前提で使うと効率的です。

### 33.3.2 Slack・チャットの要約を週報に組み込む

今週のコミュニケーションログを週報に反映する方法です。

以下は私の今週の Slack のやり取り (抜粋) です。  
このコミュニケーションをもとに、週報の「今週の動き」セクションを作成してください。  
顧客名・プロジェクト名は仮名に置き換えて構いません。

【Slack のやり取り (今週分の抜粋)】

(コピーして貼り付ける)

■ 以下の形式で出力してください：

- 主なコミュニケーション・連携事項 (箇条書き)
- 解決した課題
- 未解決の懸念点



**警告:****社内情報のセキュリティに注意**

Slack のやり取りには顧客名・金額・個人名などの機密情報が含まれることがあります。Claude に渡す前に、以下の対処を検討してください。

- 顧客名 → 「A 社」「B 社」に置き換える
- 個人名 → イニシャルや「A さん」に置き換える
- 金額・契約内容 → 「〇〇円規模」「大型案件」のように抽象化する

claude.ai の会話はデフォルトでモデルの学習に使用されませんが、社内のセキュリティポリシーとの整合を必ず確認してください。

## 33.4 報告書・提案書の下書きを作る

### 33.4.1 「80 点の下書き」を 5 分で用意する

報告書や提案書の作成でよく起きる問題は、「白紙から書き始めると 2〜3 時間かかる」ことです。Claude を使えば、**構造と素材を渡すだけで 80 点の下書き**が 5 分で出来上がります。

KDDI の minorun365 氏は、この考え方を「仕様書駆動」と呼んでいます ([Qiita 記事](#))。PLAN.md (計画)・SPEC.md (仕様)・TODO.md (タスク)・KNOWLEDGE.md (知識) の 4 文書を用意し、Claude に渡すことで、経費精算を 30 分から 5-10 分に短縮するなどの成果を上げています。プレゼン資料の自動生成にも応用しているとのこと。

あなたがすること：

1. 報告書の「目的」「背景」「伝えたいこと」を箇条書きにする (5 分)
2. Claude に渡して下書きを生成させる (1〜2 分)
3. 事実確認・数字の修正・自分の言葉への調整 (10〜15 分)

合計 20 分程度で、従来 1〜2 時間かかっていた報告書が完成します。

### 33.4.2 基本プロンプト

以下の素材をもとに、報告書の下書きを作成してください。

**【目的】**

(例：新規システム導入の承認を得る)

**【背景・現状の課題】**

(箇条書きで)

**【提案内容】**

(箇条書きで)

**【期待効果】**

(数字や具体的な変化があれば)

(次のページに続く)

(前のページからの続き)

【懸念事項・リスク】

(把握しているものを)

【読者】

(例：部長・役員会)

【文体・トーン】

(例：フォーマル、簡潔に)

【目標文字数】

(例：800～1,000 字)

### 33.4.3 トーン別に出し分ける

同じ内容でも、読者によって表現を変える必要があります。Claude は「トーン指定」だけで自動的に調整します。

#### 社内向け（チームメンバー・直属の上長宛）

以下の素材をもとに報告書を作成してください。

読者は私の直属の上長と同じチームのメンバーです。

背景知識が共有されているため専門用語を使って構いません。

率直でスピード感のある文体で。

【素材】

(上と同じ素材を貼り付ける)

#### 経営層向け（役員・社長宛）

以下の素材をもとに報告書を作成してください。

読者は役員・経営陣です。

- 結論を冒頭に置く（BLUF 形式）
- 数字と事業インパクトを強調する
- 詳細な手順・技術的説明は省く
- 「なぜ今やるのか」を明確に

【素材】

(上と同じ素材を貼り付ける)

#### 顧客向け（社外・取引先宛）

以下の素材をもとに報告書を作成してください。

読者は社外の顧客（発注側）です。

- 専門用語・社内略語を使わない
- 顧客のメリット・利益を前面に出す

(次のページに続く)

(前のページからの続き)

- 丁寧で信頼感のある文体
- 曖昧な表現を避け、具体的な数字を使う

【素材】

(上と同じ素材を貼り付ける)

### 33.4.4 複数バリエーションを出して選ぶ

「どの表現が一番伝わるかわからない」場合は、複数バリエーションを出して比較する方法が効果的です。

以下の提案書の冒頭段落を、3 パターン書いてください。

【内容】

(伝えたいことを箇条書きで)

【3 パターンの違い】

- パターン A：コスト削減を前面に出す
- パターン B：リスク回避・安全性を前面に出す
- パターン C：競合との差別化・将来性を前面に出す

読者は経営会議の参加者（役員 5 名）です。

3 パターンを見比べることで、「どの切り口が刺さるか」を判断しやすくなります。

**つまづきポイント：生成された下書きが「それっぽいが薄い」**

素材として渡す情報が少ないと、Claude は当たり障りのない文章を生成します。「具体的な数字」「固有名詞」「比較対象」を素材に含めると、一気に内容が充実します。たとえば「コストが削減できる」ではなく「月次コストが現在の 30 万円から 20 万円に削減できる」のように。

## 33.5 「振り返り」は人間がやる——これが唯一の黄金律

ここが、この記事でもっとも大切なセクションです。

### 33.5.1 AI が書く「振り返り」には意味がない

Claude は日報を上手に書きます。要約・整形・体裁の整えは AI が得意です。しかし「振り返り」だけは、AI に任せてはいけません。

日報ツール「nippon」を作った開発者 nwiizo 氏は、自身の[ブログ記事](#)でこう言っています。

「思考・努力・羞恥心まで外注してはいけない」

自動生成された振り返りを読んでも、何も残らない——そういう感覚を経験したことがある方もいるはずです。事実の要約は機械にできますが、「なぜうまくいったか」「何を感じたか」「次回どうするか」という内省は、人間にしかできません。

これは「AI に頼るな」という話ではなく、むしろ逆です。書く手間を AI に任せるからこそ、振り返りに使う時間と精神的余裕が生まれるのです。

### 33.5.2 コルブの経験学習サイクル

心理学者コルブは、経験が学びに変わるには 4 つのステップが必要だと言っています。

1. **経験する** — 何かをやってみる
2. **省察する** — 起きたことを振り返る
3. **概念化する** — 何が機能したかを理解する
4. **実験する** — 次の行動に活かす

AI が担えるのは「1」の記録整理だけです。「2～4」は人間の仕事です。

### 33.5.3 「振り返りの問い」を AI に出させる

では、振り返りはどうやればいいのか。

おすすめは「AI に問いを出させ、人間が 5 分で答える」形式です。

以下の今日の日報をもとに、私が振り返るための問いを 3 つ出してください。

【今日の日報】

(Claude が生成した日報を貼り付ける)

【条件】

- 「何をやったか」ではなく「なぜそう判断したか」「何を感じたか」を深掘りする問い
- 表面的な感想ではなく、行動・思考のパターンを発見できる問い
- 答えるのに 5 分以内で済む具体的な問い

出力例：

- 「A 社の資料修正に 2 時間かかった。想定より時間がかかった原因は何だと思いますか？」
- 「B さんへの確認依頼を今日まで後回しにした。何が障壁になっていましたか？」
- 「今日最も集中できた時間帯はいつですか？その環境や状態に共通点はありますか？」

この 3 つに答えるだけで、翌日への具体的な改善策が見えてきます。



注釈:

「省察ノート」を日報に添えるだけでいい

振り返りは立派な文章にする必要はありません。問いへの答えを 3～5 行書くだけで十分です。日報の末尾に「今日の気づき」という 1 セクションを設けて、そこに書き留める習慣にするとシンプルに続けられます。

## 33.6 コピペで使えるプロンプト集

### 33.6.1 日報（基本）

今日の作業メモをもとに日報を作成してください。

【作業メモ】  
(箇条書きで)

【フォーマット】

- 本日の業務：
- 成果・進捗：
- 課題・懸念：
- 明日の予定：
- 所感：

【条件】全体 300～400 字。事実ベースで。感想・励ましの言葉は不要。

### 33.6.2 週報

今週の作業メモをもとに週報を作成してください。

【作業メモ（月～金）】  
(各日のメモを箇条書きで)

【フォーマット】

- 今週の成果：
- 進行中のタスクと進捗：
- 課題・懸念事項：
- 来週の計画：
- 所感・学び：

【条件】全体 500～700 字。数字があれば具体的に記載。

### 33.6.3 月次報告

今月の活動をもとに月次報告書を作成してください。

【今月の主な活動・成果（箇条書き）】  
( )

【数値目標と実績】

目標：( )

実績：( )

(次のページに続く)

(前のページからの続き)

【課題と来月への改善策】

()

【フォーマット】

- 今月のハイライト (3 項目以内) :
- 目標達成状況 :
- 主な活動と成果 :
- 課題・反省点 :
- 来月の重点課題 :

【条件】 役員・マネージャー向け。800 字以内。結論を先に。

### 33.6.4 提案書の下書き

以下をもとに提案書の下書きを作成してください。

【目的】

(何を承認・実現したいか)

【背景・現状の課題】

(なぜ今これが必要か)

【提案内容】

(具体的に何をするか)

【期待効果・メリット】

(数字があれば)

【コスト・リスク】

(把握しているものを)

【読者】

(誰が読む文書か)

【構成】

1. 要旨 (結論と要点を 3 行で)
2. 背景・課題
3. 提案内容
4. 期待効果
5. リスクと対策
6. 実施スケジュール (案)

【条件】 全体 1,000～1,200 字。専門用語は最小限に。

### 33.6.5 振り返りの問いを生成する

今日の日報をもとに、私が 5 分で答えられる振り返りの問いを 3 つ出してください。

**【今日の日報】**

(日報テキストを貼り付ける)

**【条件】**

- 「何をしたか」の確認ではなく、「なぜ」「どう感じたか」「何が学べるか」を問う
- 答えは 3～5 行で済む具体的な問い
- 自分のパターンや成長に気づけるような問い

## 33.7 次のステップ

日報・週報の仕組みが整ったら、次はメールや朝のブリーフィングにも Claude を活用してみましょう。

- **Claude を AI 秘書にする (全体ガイド)** (ページ 265) — AI 秘書活用の全体像とレベル別の始め方
- **メール処理の自動化** (ページ 271) — 返信下書き・優先度分類・未返信の検出
- **朝のブリーフィングとタスク整理** (ページ 279) — 「今日やること一覧」を毎朝自動生成する
- **プロンプトの基本テクニック** (ページ 239) — Claude により正確に伝わる指示の書き方





## 第34章 AI 秘書を本格化する — MCP 連携と複数ツール統合

*Level 1 (ブラウザでコピー)* (ページ 265) や *Level 2 (Claude Desktop のコネクタ)* (ページ 265) で Claude を使い始めた方向けの、次のステップです。

Level 3 では **MCP (Model Context Protocol)** という仕組みを使って、Claude と複数のツールを直接つなぎます。ターミナルに「おはよう」と入力するだけで、Gmail・Slack・カレンダー・タスク管理ツールの情報がまとめて届く——そんな「本格的な AI 秘書」が実現します。



警告:

この記事はターミナルを使える方向けです

MCP の設定には、ターミナル（コマンドを入力する画面）の基本操作と、簡単な設定ファイルの編集が必要です。Claude Code のセットアップがまだの方は、まず [Claude Code をはじめる](#) を読んでから戻ってきてください。

### 34.1 Level 1・2 の限界と Level 3 で変わること

#### 34.1.1 コピペの壁

Level 1・2 の使い方には、どうしても「手作業」が残ります。

- Gmail を開いてメールをコピー → Claude に貼り付け → 返信案をコピーして Gmail に戻る
- Slack を開いてメンションをコピー → タスクリストに追加
- カレンダーを確認 → 別ウィンドウで Claude に伝える

一つひとつは小さな手間ですが、毎朝これを繰り返すと、情報収集だけで 20〜30 分かかってしまいます。また、情報源が増えるほど、この手間は積み重なっていきます。

#### 34.1.2 Level 3 で何が変わるか

MCP を使うと、Claude が外部ツールを**直接読み書き**できるようになります。

```
ターミナルに入力するだけ
$ claude
> おはよう

Claude が自動的に以下を実行:
1. Gmail を開いて未返信メールを抽出
```

(次のページに続く)

(前のページからの続き)

- # 2. Slack でメンションされたスレッドを確認
- # 3. 今日のカレンダーを取得
- # 4. Todoist からタスクリストを読み込む
- # 5. 優先度付きのブリーフィングを生成・保存

手動でコピペしていた作業が、コマンド一つで完結します。Art of Science のニュースレターでは「Claude Code はコーディングツールではない。人生の自動化インフラだ」と表現されています——支出分析で月 \$400 の節約を発見したり、12,000 枚の写真を 20 分で整理したりと、コード以外のタスクにこそ威力を発揮します。

## 34.2 MCP とは何か

MCP (Model Context Protocol) は「Claude と外部ツールをつなぐ翻訳者」です。

イメージとしては、Claude という賢い秘書がいるが、外の世界 (Gmail、Slack、GitHub など) の言葉を理解できない——そこで MCP サーバーという「通訳」を間に置くことで、Claude が各サービスと会話できるようになります。

```
Claude ←→ MCP サーバー (Gmail 用) ←→ Gmail
Claude ←→ MCP サーバー (Slack 用) ←→ Slack
Claude ←→ MCP サーバー (Todoist 用) ←→ Todoist
```

各サービスに対応した「MCP サーバー」というプログラムを設定ファイルに登録することで、Claude がそのサービスを操作できるようになります。

**技術的な補足 (エンジニア向け) :** MCP は Anthropic が策定したオープンプロトコルで、JSON-RPC ベースの通信仕様です。ツール (tool)、リソース (resource)、プロンプト (prompt) の 3 種類の機能を Claude に提供できます。設定は `claude_desktop_config.json` または Claude Code の `~/.claude/settings.json`で行います。



注釈:

### MCP サーバーはサードパーティも公開しています

Gmail、Slack、GitHub など主要サービスの MCP サーバーは、公式または有志による実装が GitHub に公開されています。多くの場合、`npm install`などのコマンド一行で導入できます。

## 34.3 実践者のアーキテクチャ

実際に MCP を使って AI 秘書を構築した方々の事例を紹介します。それぞれの状況・課題・解決策が参考になるはずです。

### 34.3.1 ired : 「おはよう」で5サービス統合ブリーフィング

**状況:** 執行役員として多くの情報源を毎朝チェックする必要があった

アイレット 後藤氏は、Claude Code と MCP を組み合わせて5つの情報源（メール・Slack・カレンダー・タスク管理・ノート）を統合したワークフローを構築しました。

**構成:**

- **Gmail MCP** — 2 営業日以上未返信のメールを自動検出
- **Slack MCP** — メンションされたスレッドを抽出
- **Google Calendar MCP** — 今日・明日の予定を整理
- **Todoist MCP** — タスクの優先度を再評価
- **Obsidian** — ブリーフィング結果を日次ノートとして保存

ターミナルで おはよう と入力すると、上記5サービスを横断した優先度付きブリーフィング（赤/黄/緑の3段階で分類）が自動生成・保存されます。朝のタスク整理が **30 分から 3 分**に短縮されました（ired.media 記事より）。

### 34.3.2 クラスメソッド : GTD と Zettelkasten の統合

**状況:** タスク管理（GTD）と知識管理（Zettelkasten）が分断されており、情報の再利用が難しかった

クラスメソッドの「きんじょー」氏は、Claude Code をハブとして両方の管理システムを統合するワークフローを構築しました。

**カスタムコマンド:**

```
/morning # 朝のプランニング。今日やることをインボックスから整理
/inbox # 未処理の項目を整理してタスクまたはナレッジに振り分け
/review # 週次レビュー。完了タスクの振り返りとナレッジの更新
```

この仕組みにより、情報を「集める → 整理する → 考える」という作業が「考える → 判断する」へとシフトしました。Claude が整理を担うことで、人間は意思決定に集中できるようになったということです（クラスメソッド記事より）。

### 34.3.3 TechRacho : 38 プロジェクトの並行管理

**状況:** 中間管理職として38プロジェクトを並行管理しており、コンテキストスイッチが激しかった

morimorihoge 氏は、GitHub・GitLab・Backlog・Slack・DocBase という5つのプロジェクト管理ツールを Claude Code で横断管理するワークフローを構築しました。

**中心となるコマンド:**

```
/project-update all
```

このコマンドを実行すると、全プロジェクトの更新情報（プルリクエスト、課題、Slack スレッド）が一元整理され、対応が必要な事項が優先度付きでリストアップされます。各プロジェクトのコンテキストを

頭に入れ直す時間が大幅に削減され、コンテキストスイッチのストレスが軽減されたと報告しています (TechRacho 記事より)。

**つまづきポイント：プロジェクト数が増えすぎてコマンドが遅くなる**

project-update all のように全プロジェクトを一括更新すると、API コールが多くなり処理に時間がかかることがあります。対策として、①頻繁に更新されるプロジェクトとそうでないプロジェクトを分けてコマンドを作る、②週次・日次でチェックするプロジェクトを分ける、といった工夫が有効です。

**34.3.4 Blattman : 9 スキル体制のエグゼクティブアシスタント**

**状況:** 研究者・著者として膨大なメールと情報を管理する必要があり、5,000 通以上の未読メールを抱えていた

Chris Blattman 氏は、9 つの専門スキルを持つ AI エグゼクティブアシスタントを構築しました。各スキルが担当領域を持ち、Claude が適切なスキルを呼び出して処理します。

**9 つのスキル体制:**

- 1. メール優先度分類
- 2. 未読メールのサマリー
- 3. 返信下書き生成
- 4. 会議スケジューリング
- 5. 研究論文の要約
- 6. 原稿フィードバック
- 7. 学生メール対応
- 8. タスクリスト管理
- 9. 週次レビュー

このシステムにより、5,000 通の未読メールを重要な 6 通まで絞り込み、週 1〜2 時間の時短を実現しました。注目すべきは「段階的自律性」という設計思想——次のセクションで詳しく説明します (claudeblattman.com 記事より)。

**34.4 よく使われる MCP サーバー**

自分のワークフローに合わせて、必要な MCP サーバーを選んでください。

サービス	用途	入手方法
Gmail / Google Calendar	メール・予定の読み書き	サードパーティ実装 (@postlab/google-workspace-mcp 等)
Slack	メッセージ送信・チャンネル読み取り	@modelcontextprotocol/server-slack

次のページに続く

表 1 – 前のページからの続き

サービス	用途	入手方法
GitHub	リポジトリ・Issue・PR の管理	@modelcontextprotocol/server-github
Notion	ページ・データベースの読み書き	@notionhq/notion-mcp-server (Notion 公式)
Todoist	タスクの作成・更新・完了	サードパーティ実装あり
Obsidian	Markdown ノートの読み書き	コミュニティ実装あり
ファイルシステム	ローカルファイルの読み書き	@modelcontextprotocol/server-fileSystem



注釈:

#### MCP サーバーの最新情報は公式ページで確認

MCP サーバーのエコシステムは急速に発展しています。利用可能なサーバーの最新一覧は [Anthropic の公式 MCP ページ](#) や [GitHub](#) を確認してください。

## 34.5 段階的自律性の設計

Blattman 氏が提唱する「段階的自律性」は、AI 秘書を安全に育てるための重要な考え方です。

### 34.5.1 3つの段階

#### 段階 1：提案 (Suggest)

Claude がアクションを提案し、人間が承認してから実行します。

Claude: 「田中さんへの返信として、以下の下書きを作りました。送信しますか？」

[下書き本文...]

(y) 送信 / (e) 編集 / (n) キャンセル」

新しい MCP サーバーを導入した直後は、必ずこの段階から始めることをおすすめします。Claude がどんな判断をするかを確認できるからです。

#### 段階 2：承認 (Approve)

定型的なアクションは自動で実行し、重要なアクションのみ確認を求めます。

# ポリシーファイルの例

- Slack のスタンプ反応: 自動実行 OK
- 社内メールの返信下書き: 確認が必要
- 外部へのメール送信: 必ず確認する
- カレンダーへの予定追加: 確認が必要
- 予定の変更・削除: 必ず確認する

### 段階 3：自動化 (Automate)

信頼性が確認できたアクションは、完全に自動化します。

- # 完全自動化できる例
  - 毎朝のブリーフィング生成（読み取りのみ）
  - 日報の下書き作成（Obsidian への保存）
  - 週次タスクレビューの実行

#### 重要：書き込み操作は慎重に

メールの送信、タスクの削除、カレンダーの変更など、外部に影響を与える「書き込み操作」は、十分な信頼が積み上がるまで自動化しないことをおすすめします。読み取りや下書き生成から始め、実際の送信・変更は人間が確認するフローを維持しましょう。

## 34.6 ポリシーファイルで AI の行動を制御する

段階的自律性を実装する具体的な方法が、**ポリシーファイル**です。Claude Code では `CLAUDE.md` に AI の行動ルールを記述できます。

### 34.6.1 ポリシーファイルの例

#### # AI 秘書 ポリシーファイル

##### ## 自動実行してよいこと（確認不要）

- メール・Slack・カレンダーの読み取り
- Obsidian へのノート保存
- タスクリストの読み取りと優先度評価
- 朝のブリーフィング生成

##### ## 確認が必要なこと

- メール返信下書きを実際に送信する前
- タスクの完了マーク（間違いのリスクがあるため）
- カレンダーへの予定追加・変更

##### ## 絶対にしてはいけないこと

- 「全員に返信」での送信
- 社外メールの自動送信
- 予定の削除（変更のみ可）
- 機密ファイルを外部サービスにアップロード

##### ## 判断に迷ったとき

- 人間関係が重要なメールは人間が書く
- 初めての相手へのメールは確認を求める
- 緊急フラグがついた事項は即座に通知する

このポリシーファイルを Claude Code の `CLAUDE.md` に含めることで、Claude が一貫した判断基準で動くようになります。



注釈:

#### ポリシーは育てるもの

最初から完璧なポリシーを作ろうとしないこと。使いながら「ここは自動化できる」「ここは確認が必要だった」という経験を積み、少しずつポリシーを更新していきましょう。

## 34.7 セキュリティとプライバシー

複数のツールを統合することは便利ですが、セキュリティ上の注意点もあります。

### 34.7.1 MCP 接続に関するリスク

#### 認証情報の管理

MCP サーバーの設定には、各サービスの API キーやトークンが必要です。これらは設定ファイルに記述しますが、**絶対に Git にコミットしてはいけません**。`.env` ファイルや専用のシークレット管理ツールを使いましょう。

#### アクセス権限の最小化

OAuth でサービスと接続する際は、**最小限の権限のみ付与**してください。

- ✓ Gmail: 読み取りのみ（まず読み取りから始める）
- ✓ Slack: 自分が所属するチャンネルのみ
- × 全メールボックスへのフルアクセス（最初から不要）

#### 会社のポリシー確認

個人の Gmail・Slack であれば比較的自由に試せますが、**会社のアカウントを接続する前に IT・セキュリティ部門への確認**が必須です。多くの組織では外部サービスとの API 連携に関するポリシーがあります。

### 34.7.2 プライバシーに関する注意



警告:

#### Claude Code はローカル実行です

Claude Code は自分のパソコンで動きます。MCP サーバーを通じてメール・Slack 等を読み込む場合、そのデータは Claude の API を通じて処理されます。機密情報（顧客データ、個人情報、財務情報）を扱う場合は、Anthropic のプライバシーポリシーと自社のセキュリティポリシーを確認した上で判断してください。

## 34.8 始め方のロードマップ

「すべてを一気にやらない」——これが最も大切なメッセージです。

実際に複数ツールの統合を試みた人の多くが、「設定が複雑で途中で挫折した」「動かなくて諦めた」という経験をしています。1つの MCP サーバーから始め、確実に動作を確認してから次に進むことが成功の鍵です。

### 34.8.1 Week 1：ファイルシステム MCP で基礎を学ぶ

最初に試すべきは、**ファイルシステム MCP** です。認証情報が不要で、自分のパソコンのファイルを読み書きするだけなので、失敗しても影響がありません。

```
Claude Code に以下を設定 (~/.claude/settings.json)
{
 "mcpServers": {
 "filesystem": {
 "command": "npx",
 "args": [
 "-y",
 "@modelcontextprotocol/server-filesystem",
 "/Users/yourname/Documents" # アクセスを許可するフォルダ
]
 }
 }
}
```

設定後、**claude** を起動して「Documents フォルダのファイル一覧を見せて」と入力してみましょう。これが動けば、MCP の基本的な仕組みが理解できます。

### 34.8.2 Week 2：カレンダー MCP で読み取りを体験する

次は **Google Calendar MCP** です。読み取りのみの設定から始め、「今週の予定を整理して」と Claude に依頼してみましょう。

読み取りが安定したら、毎朝の予定確認を Claude に任せる習慣を作ります。

### 34.8.3 Week 3：タスク管理ツールを接続する

Todoist または Notion など、自分が使っているタスク管理ツールの MCP を追加します。

カレンダーとタスクリストが統合されると、「今日やること」の整理が一気に楽になります。

### 34.8.4 Week 4 以降：メール・Slack の統合

メールと Slack は最も効果が大きいですが、その分リスクも大きいです。**最初は読み取りのみ**で接続し、返信の送信は必ず手動で確認するフローを維持しましょう。



## ✓ 最初のゴール設定例

「毎朝ターミナルで `/morning` を実行すると、今日の予定・未返信メール・Slack メンションが一覧で確認できる状態を作る」

## つまづきポイント：MCP サーバーが動かない

MCP の設定で最もよくあるエラーが「MCP サーバーが起動しない」です。原因の多くは以下のどれかです。① `node` や `npx` のバージョンが古い (`node --version` で確認)、②設定ファイルの JSON 記法のミス（カンマ漏れなど）、③ API キー・トークンの誤り。まずターミナルで直接 `npx @modelcontextprotocol/server-filesystem` を実行してエラーメッセージを確認してください。

## 34.9 まとめ：AI 秘書の完成形

Level 3 まで構築すると、こんな一日が実現します。

朝:

```
$ claude
```

```
> おはよう
```

→ 今日の予定（3 件）・未返信メール（5 通）・Slack メンション（2 件）が優先度付きで整理されブリーフィング完了（所要時間：3 分）

日中:

```
> 田中さんへの返信下書きを作って
```

→ メール本文の文脈を踏まえた返信案が生成。確認後に送信

夕方:

```
> 今日の日報を作って
```

→ カレンダーの予定と Todoist の完了タスクをもとに日報下書きが生成

この状態に至るまで、多くの人は数週間～1 ヶ月かけています。焦らず、1 週間ごとに 1 つの MCP サーバーを追加するペースで進めてください。

## 34.10 次のステップ

- Claude Code をはじめる — Claude Code のインストールと初期設定
- 朝のブリーフィングとタスク整理 (ページ 279) — Level 2 (Claude Desktop) での朝のワークフロー構築
- 日報・週報・報告書の自動化 (ページ 287) — 日報生成を MCP と組み合わせる方法

- *Claude* を AI 秘書にする (入門) (ページ 265) — AI 秘書全体像とレベル別の始め方

## 第35章 Claude で PDF・ファイルを活用する

取引先から届いた英語の契約書、山積みになった市場調査レポート、会議のたびに更新される仕様書——ドキュメントを読んで理解することに、毎日どれだけの時間を使っていますか。

Claude にファイルを渡せば、「要点を3行でまとめて」「この条項は何を意味する?」「日本語に訳して」と話しかけるだけで答えが返ってきます。読む・探す・翻訳するという作業を Claude に任せることで、本来の判断や意思決定に時間を使えるようになります。

この記事では、Claude にファイルを渡して業務に活かすための基本を解説します。

### 35.1 Claude にファイルを渡すとできること

ファイルを添付して Claude に質問することで、次のような作業が格段に楽になります。

**報告書・資料の要約** 100 ページを超える調査報告書も、「エグゼクティブサマリーとして5点にまとめて」と指示するだけで整理されます。会議前の資料確認が数分で終わります。

**情報の抽出・整理** 「この契約書から支払い条件と解約条項だけを抜き出して表にして」のように、特定の情報だけを取り出すことが得意です。複数の資料を横断して比較することもできます。

**英語・外国語ドキュメントの翻訳** 英語の論文、海外サプライヤーからの仕様書、法的文書——専門用語を含む文書でも、文脈を理解した自然な翻訳が得られます。

**データの読み取りと分析** グラフや表を含む PDF でも、数字を読み取って傾向を説明したり、特定の値を探したりできます。Claude は画像として埋め込まれたグラフも認識できます。



注釈:

#### 無料プランでも PDF・ファイルの添付は使えます

claude.ai の無料アカウントでも、ファイルのアップロードと分析は利用できます。1日あたりのメッセージ数に制限はありますが、まずは試してみましょう。

### 35.2 ファイルをアップロードする手順

claude.ai のブラウザ版でファイルを添付する方法を説明します。

#### ステップ 1: チャット画面を開く

claude.ai にログインし、新しい会話を開始します。

#### ステップ 2: ファイル添付ボタンをクリックする

入力欄の左側にあるクリップアイコン (+ボタン) をクリックすると、ファイル添付メニューが開きます。

ステップ 3: ファイルを選んで質問する

「Upload from computer（コンピューターからアップロード）」を選択して、読み込みたいファイルを選びます。ファイルがアップロードされたら、入力欄にやりたいことを入力して送信します。

(PDF を添付した後)  
このレポートの要点を箇条書きで 5 つ教えてください。

つまづきポイント：ファイルを添付したのに「読めない」と言われる

ファイルを添付した後、何も指示を入力せずに送信しても Claude はどう応答すればよいかわかりません。「添付できた」と思ったら、必ず一緒に質問や指示を入力してから送信しましょう。

35.3 対応形式と制限

Claude が読み取れるファイルの種類と、制限事項を確認しておきましょう。

ファイル形式	主な用途
PDF（パスワードなし）	報告書、論文、契約書、マニュアルなど
Word（.docx）	ビジネス文書、提案書、議事メモなど
Excel（.xlsx）	表データ、集計結果（有料プランで直接読み込み可）
画像（PNG, JPEG, GIF, WebP）	スクリーンショット、図表、写真の読み取り
テキスト・その他（.txt, .csv, .html, .json）	メモ、データ、ログなど

制限事項

項目	上限
1 ファイルあたりのサイズ	30MB
PDF の最大ページ数	100 ページ



警告:

**パスワード保護・暗号化された PDF は読み取れません**

セキュリティのためにパスワードがかかっている PDF は、Claude に渡す前にパスワードを解除する必要があります。また、スキャンした書類をそのまま PDF 化したものは、文字データが含まれないため読み取り精度が下がる場合があります。

35.4 Claude が PDF を読む仕組み

Claude の PDF 処理には、他のツールと異なる特徴があります。

PDF を受け取ると、Claude は**テキスト抽出と画像解析を同時に行います**。各ページを画像として認識しながら、そこに含まれる文字情報も取得します。この二段構えの処理により、文章だけでなく、グラフの傾向、表のレイアウト、図の内容なども理解できます。

たとえば「売上グラフが右肩上がりになっているのはどの四半期から？」のような、画像に関する質問に

も回答できます。

## 35.5 PDF 活用の基本パターン

### 35.5.1 要約する

長い文書の内容を、目的に合った形でまとめてもらいます。「何が書いてあるか把握したい」というときに使います。

**プロンプト例：**

添付の報告書の要点を箇条書きで5つ教えてください。  
技術的な詳細より、意思決定に必要な情報を優先してください。

「要約して」だけで送るよりも、目的や読む人の立場を一言添えるだけで、返ってくる内容が大きく変わります。会議前の資料確認、上司への報告用まとめなど、シーン別のプロンプトテンプレートは[要約の詳細ガイド](#) (ページ 315) で紹介しています。

### 35.5.2 情報を抽出する

文書の中から特定の情報だけを取り出して整理してもらいます。複数の書類を比較するときにも使えます。

**プロンプト例：**

この契約書から、契約期間・支払い条件・解約条件・違約金の規定を  
抜き出して表形式でまとめてください。該当ページ番号も記載してください。

「ページ番号も記載して」と指示すると、後で元の文書で確認しやすくなります。

### 35.5.3 翻訳する

英語をはじめとする外国語の文書を日本語に訳してもらいます。専門用語の多い文書でも、文脈を理解した翻訳が得られます。

**プロンプト例：**

この英語の仕様書を日本語に翻訳してください。  
専門用語はカタカナ表記で、ビジネス文書として自然な文体をお願いします。

分野別（ビジネス文書・論文・契約書・技術仕様書）のプロンプトテンプレートは[翻訳の詳細ガイド](#) (ページ 337) で紹介しています。

## 35.6 うまく読み取れないときは

PDF の読み込みでよくあるトラブルは、次の3つに集約されます。

- **スキャン PDF で精度が低い** — 画像データのため文字認識が不完全になることがある
- **100 ページを超える PDF** — 1 リクエスト 100 ページまでの上限がある。章ごとに分割して送る

- ・ **パスワード保護付き PDF** — 事前にパスワードを解除してから送る必要がある

それぞれの具体的な対処手順は、[PDF が読み込めないときの対処法](#) (ページ 323) で詳しく解説しています。

## 35.7 Projects で繰り返し使う

同じ資料を何度も参照する場合や、チーム全員で共通の資料を使いたい場合は、**Projects** (プロジェクト) 機能が便利です。

Projects では、PDF などのファイルを「ナレッジ (知識の源)」として登録しておけます。登録されたファイルは、そのプロジェクト内のすべての会話で自動的に参照されます。

たとえば、自社の製品仕様書やよく参照する法令をプロジェクトに登録しておけば、毎回ファイルを添付しなくても「この製品のスペックは?」「この規定の解釈は?」と聞くだけで回答が得られます。

**使い方の流れ：**

1. claude.ai の左サイドバーから「New project (新しいプロジェクト)」を作成する
2. プロジェクト設定画面で「Add content (コンテンツを追加)」からファイルをアップロードする
3. 以降、そのプロジェクト内での会話ではファイルが自動参照される



注釈:

### Projects は無料プランでも使えます

Projects の基本機能 (ファイルの登録・参照) は、無料プランでも利用できます。登録できるファイル数やサイズに制限があることがあるため、詳しくは[プランの比較](#) (ページ 15) を参照してください。

## 35.8 もっと詳しく

PDF・ファイル活用のより深いテクニックは、以下のページで解説しています。

### 35.8.1 PDF を要約する方法とプロンプト例 (ページ 315)

会議前の資料確認、経営層への報告用まとめ、契約書の確認ポイント整理など、シーン別のプロンプトテンプレートと長文 PDF への対処法を解説します。

### 35.8.2 PDF を翻訳する方法 (ページ 337)

ビジネス文書・学術論文・契約書・技術仕様書の分野別テンプレートと、翻訳精度を上げるテクニックを解説します。

### 35.8.3 ファイルを出力・生成する方法 (ページ 331)

Claude に作らせた文書を PDF・Word・CSV などの形式で保存する方法。Artifacts 機能の使い方と実用例を解説します。

### 35.8.4 PDF が読み込めないときの対処法 (ページ 323)

ファイルサイズ超過、パスワード保護、スキャン PDF の精度問題など、よくあるトラブルと具体的な解決手順をまとめています。

## 35.9 よくある質問

### 35.9.1 複数のファイルを一度に送ることはできますか？

はい、1 回のメッセージに複数のファイルを添付できます。「A と B のファイルを比較して」のように複数ファイルを一緒に渡して分析させることも可能です。ただし、合計サイズが 30MB 以内に収まる必要があります。

### 35.9.2 送ったファイルの内容は Anthropic に保存されますか？

Anthropic の方針では、claude.ai での会話はデフォルトでモデルの学習には使用されません。ただし、サーバーには一時的に保存されます。機密性の高いファイルを送る場合は、社内のセキュリティポリシーを確認した上で使用してください。顧客名や金額など特に機密性の高い情報は、仮名やダミーデータに置き換えてから使うことをおすすめします。

### 35.9.3 Word や Excel のファイルは送れますか？

はい、どちらも claude.ai に直接アップロードできます。Word (.docx) はそのまま添付して質問できます。Excel (.xlsx) は、アカウント設定で「コード実行とファイル作成 (Analysis Tool)」を有効にすると読み込めるようになります (有料プラン)。無料プランの場合は、Excel を CSV として書き出してから送る方法が使えます。

### 35.9.4 PDF の特定のページだけを読ませることはできますか？

現時点では、特定ページだけを選んで送る機能はありません。「10～15 ページの内容について教えて」のように、ページ番号を指定して質問することで、Claude がそのページに集中して回答してくれます。あるいは、Adobe Acrobat などのアプリを使って必要なページだけを別ファイルとして書き出してから送る方法もあります。

## 35.10 次のステップ

- **プロンプトの基本テクニック** (ページ 239) — より的確な指示の書き方を学ぶ
- **Claude Web の使い方** (ページ 89) — Projects やその他の Web 版機能を活用する
- **AI 秘書として使う** (ページ 265) — ファイル活用を含めた業務自動化の全体像





---

## 第36章 Claude で PDF を要約する方法とプロンプト例

「読まなきゃいけない資料」がたまっていないですか。

市場調査レポート、競合他社の発表資料、会議の事前配布資料、英語の技術論文——1 つひとつの内容は重要でも、すべてをじっくり読む時間はなかなかありません。

Claude に PDF を渡して要約を頼むとき、「要約して」の一言だけで送るのと、目的を伝えてから送るのは、返ってくる内容がまったく違います。この記事では、シーン別のプロンプトテンプレートと、長文 PDF への対処法を解説します。



注釈:

### この記事の前提

PDF のアップロード手順や対応形式については、[Claude で PDF・ファイルを活用する](#) (ページ 309) で解説しています。ここでは「どう要約を指示するか」に絞って解説します。

### 36.1 要約は「短くすること」ではない

「要約して」と伝えと、Claude は全体をバランスよく短くまとめようとします。しかし、あなたが本当に知りたいのは「全体の圧縮版」ではないはずです。

- 明日の会議前に確認したいなら → **決定事項と自分への影響**
- 上司に報告するなら → **意思決定に使える数字と推薦事項**
- 競合資料を読むなら → **自社との差分と注目点**
- 契約書を確認するなら → **義務、リスク、期限**

同じ PDF でも、目的が違えば必要な情報は変わります。Claude にとって、「何のために要約するのか」は出力の方向を決める最も重要な情報です。これを伝えるだけで、要約の使い勝手が大きく変わります。

### 36.2 シーン別プロンプトテンプレート

#### 36.2.1 会議前の資料確認

**使うシーン：** 会議まで 30 分しかない。配布された資料を読む時間はないが、最低限の内容は把握しておきたい。

**プロンプト：**

(PDF を添付)

この資料を会議前に把握するために読んでいます。時間が 10 分しかありません。

以下の形式で教えてください：

- この資料が伝えたいこと (1~2 文)
- 会議で議論になりそうなポイント (3 つ以内)
- 事前に自分で確認しておくべき数字・事実 (あれば)

**なぜこう書くのか：**「時間が 10 分しかない」という制約を伝えることで、Claude が情報を絞り込む基準を持てます。「議論になりそうなポイント」を求めることで、単なる要点リストではなく会議で使える情報が得られます。

### 36.2.2 上司・経営層への報告用まとめ

**使うシーン：** 調査結果や外部レポートの内容を、上司や経営層に口頭で報告しなければならない。

**プロンプト：**

(PDF を添付)

このレポートの内容を、経営層への口頭報告 (5 分程度) に使えるようにまとめてください。

以下の構成でお願いします：

1. 結論・提言 (2~3 行)
2. それを支える主な根拠 (3 点)
3. 我々のビジネスへの示唆 (具体的に)
4. 懸念点・リスク (あれば)

技術的な詳細より、判断・意思決定に必要な情報を優先してください。

数字が出てくる場合は省略せず具体的に記載してください。

**なぜこう書くのか：**「経営層への報告」という読み手の文脈を与えると、Claude は言葉選びをその読者に合わせます。「判断・意思決定に必要な情報を優先して」と明示することで、背景説明より実用的な示唆が前に出てきます。

### 36.2.3 英語論文・レポートの要約 (日本語で)

**使うシーン：** 英語の論文や海外レポートが送られてきたが、読み込む時間も余裕もない。まず日本語で概要をつかみたい。

**プロンプト：**

(英語の PDF を添付)

この英語のレポートを日本語で要約してください。

以下の点を含めてください：

- 研究・調査の目的と対象
- 主な発見事項 (重要な数値データを含めて)

(次のページに続く)

(前のページからの続き)

- 結論と著者らの主張
- 日本のビジネス環境への適用可能性（あれば）

専門用語はわかりやすく補足しながら訳してください。

元の文書のどのセクションで語られているかも合わせて記載してください。

**なぜこう書くのか：** 含めてほしい要素を列挙することで、重要な情報が抜け落ちにくくなります。「元のセクション名を記載して」を加えると、気になった部分を後から原文で読み直しやすくなります。



注釈：

#### 翻訳と要約は一度にまとめて頼める

「まず全文翻訳して、そのあと要約して」と2回に分けて頼む必要はありません。「日本語で要約してください。重要な箇所は詳しく」のようにまとめて指示するほうが、流れが自然で結果も一貫性が出ます。

### 36.2.4 契約書・法的文書の確認ポイント整理

**使うシーン：** 取引先から送られてきた契約書を確認しなければならないが、法律の専門家ではないので何に気をつければいいのか分からない。

**プロンプト：**

（契約書の PDF を添付）

この契約書を確認しています。法律の専門家ではないため、重要な点を分かりやすく整理してください。

以下の観点で確認してください：

- 契約期間と更新・解約の条件（特に自動更新の有無）
- 自社の主な義務と責任範囲
- 相手方の主な義務と責任範囲
- 料金・支払い条件（金額、時期、方法）
- ペナルティ・違約金の規定
- 注意が必要と思われる条項

一般的なビジネス契約と比べて、特殊・不利と思われる点があればその旨も教えてください。

**なぜこう書くのか：** 「法律の専門家ではない」と伝えると、Claude は専門用語を噛み砕いて説明してくれます。「一般的な契約と比べて特殊な点」を聞くことで、見落としがちなリスクが浮かび上がってきます。



警告：

#### 契約書の最終確認は専門家へ

Claude は契約書の読み解きを助けてくれますが、法的なアドバイスを提供するものではありません。重要な契約の締結前には、弁護士など法律の専門家に確認することをおすすめします。

### 36.2.5 複数資料の横断比較

**使うシーン：** 複数の提案書や競合製品の仕様書など、2〜3 つのファイルを並べて比較したい。

**プロンプト：**

(複数の PDF を添付)  
添付した 3 社の提案書を比較してください。

以下の項目について、各社の内容を並べた比較表を作ってください：

- 提供サービスの内容
- 価格・費用（初期費用・月額・オプション）
- 導入期間の目安
- サポート体制
- 強みと弱み（各社 1〜2 点ずつ）

最後に、コストパフォーマンスと導入の手軽さの観点から、あなたのおすすめを教えてください（判断根拠も一緒に）。

**なぜこう書くのか：** 比較する項目を先に列挙することで、3 社が同じ軸で並んだ表が得られます。「おすすめを聞く」ことで、表を読んで自分で考えるひと手間を省けます（最終判断はもちろん自分で行います）。



注釈:

#### 複数ファイルを一度に添付できる

claude.ai では 1 回のメッセージに複数のファイルを添付できます。合計サイズが 30MB 以内であれば、3〜4 ファイルをまとめて渡して比較させることが可能です。

## 36.3 長い PDF（100 ページ超）を要約するコツ

Claude の 1 回のやり取りで処理できる PDF は最大 100 ページまでです。それより長い資料を扱うときは、次の方法が有効です。

### 36.3.1 目次を先に読ませて、必要な章を絞る

100 ページを超える資料でも、最初から最後まで読む必要があるケースは多くありません。まず目次や概要だけを送って「どの章が自分の目的に関係するか」を確認し、その章だけを PDF から切り出して送る方法が効率的です。

(目次ページのための PDF またはテキストを送る)  
このレポートの目次です。私は「第 3 章：市場動向の分析」と  
「第 6 章：今後の課題」だけを読みたいと思っています。  
これらの章は、この資料全体の中でどのような位置付けですか？  
他に優先して読むべき章があれば教えてください。

### 36.3.2 章ごとに分けて複数回に送る

目次を確認したら、重要な章を PDF から切り出して 1 章ずつ送る方法があります。各章の要約を積み上げてから最後に全体まとめを作るイメージです。

(第 3 章のみの PDF を送る)

先ほどの資料の第 3 章です。この章の要点をまとめてください。  
前の章では〇〇という課題が挙がっていましたが、  
この章ではその問題にどう触れられていますか？

### 36.3.3 段階的に深掘りする

全体像をつかんでから気になる部分を詳しく聞く「段階的要約」も有効です。

1. **第 1 ステップ：全体の骨格をつかむ**「この資料の章立てと、各章で言いたいことを 1 行ずつまとめてください」
2. **第 2 ステップ：関心のある章を深掘り**「第 4 章について、もっと詳しく教えてください。特に〇〇の部分が気になっています」



注釈:

#### PDF の特定ページを切り出す方法

Mac では「プレビュー」アプリで PDF を開き、サムネイル表示から必要なページだけを選択して「ファイル」→「選択したページを書き出す」で切り出せます。  
Windows ではブラウザで PDF を開き、印刷ダイアログで「ページ範囲」を指定して PDF として保存する方法が使えます。

## 36.4 要約の精度を上げるテクニック

### 36.4.1 読む人の立場と目的を最初に伝える

同じ資料でも、誰が何のために読むかによって必要な情報は変わります。Claude に読み手のプロフィールを伝えと、その人に合った要約になります。

私はこの会社への投資を検討している個人投資家です。

この年次報告書から、財務的な健全性と今後の成長可能性に絞って教えてください。

### 36.4.2 出力形式を具体的に指定する

「まとめてください」だけだと、Claude は自分で形式を決めます。使い道が決まっているなら、形式も指定したほうが後の作業が楽になります。

#### 【例 1】会議で使うスライドの素材として

- 箇条書きで 5 点
- 各点は 1 行以内
- 数字や固有名詞は省略しない

### 【例 2】上司へのメール用

- 3 段落構成（状況 → 課題 → 推薦事項）
- 1 段落あたり 3～4 文
- メール本文として使えるよう敬体で書く

### 36.4.3 「省略しないで」は逆効果なことが多い

「省略せずにすべて書いてください」という指示は、かえって返ってくる文量が増えすぎて使いにくくなる場合があります。

「省略しないで」が必要なのは「数字」「固有名詞」「条件」など、欠けると困る具体的な情報に限定するのがコツです。

#### 【NG】全体を長くしてしまう

すべての内容を省略せずに詳しくまとめてください。

#### 【OK】欠けると困る部分だけ指定する

要約の文量は自由ですが、金額・日付・企業名は省略せず正確に記載してください。

## 36.5 つまづきポイント

### 要約が浅くて「知ってること」しか書かれない

対処法: 「この文書から初めて知った情報だけを教えてください」と指示する方法が有効です。また「私はこの分野に詳しくありません。専門的な内容も噛み砕いて説明してください」と前置きすると、Claude が背景知識の説明を加えてくれます。

### 要約が長すぎて、どこが重要かわからない

対処法: 「最も重要な点を 3 つに絞ってください」のように、点数を指定して絞り込みます。あるいは「50 字以内で一番重要なことを 1 文で教えてください」と極端に短い制約をかけることで、Claude が優先順位を付けやすくなります。

### 的外れな要約が返ってくる（自分が知りたいことと違う）

対処法: 「私が本当に知りたいのは〇〇です。この文書はそれに答えていますか？」と聞き直すと、Claude が文書の内容と質問のズレを教えてくれることがあります。また、前の要約結果を引用して「この部分の◇◇について、もっと詳しく教えてください」と深掘りする方法も有効です。

### 同じ文書を別の目的で読み直したい

対処法: Claude との同じ会話の中で「今度は〇〇の観点から同じ文書を見直してください」と伝え、再アップロードなしで別の角度からの要約が得られます。

## 36.6 次のステップ

- *Claude* で PDF・ファイルを活用する (ページ 309) — アップロード手順と PDF 活用の全体像
- *Claude* で PDF を翻訳する方法 (ページ 337) — 英語論文や契約書を日本語に翻訳するテクニック
- プロンプトの基本テクニック (ページ 239) — 要約以外の場面でも使えるプロンプトの考え方





# 第37章 Claude で PDF が読み込めないときの対処法

最終更新: 2026-04-09

「PDF を送ったのにエラーになった」「アップロードはできたのに読み取りが不完全」「100 ページを超えるファイルはどうすれば?」——Claude で PDF を使い始めると、こうした壁に当たることがあります。

ほとんどの場合、原因は次の 3 つのどれかです。

- 1. ファイルサイズ・ページ数が制限を超えている（最もよくある）
- 2. PDF にパスワードがかかっている
- 3. スキャンした画像データで、文字情報がない

この記事では、各ケースの原因と具体的な解決手順を説明します。

## 37.1 まず確認するチェックリスト

問題の切り分けに迷ったら、次の 5 項目を順番に確認してみてください。

確認項目	確認方法	目安・上限
ファイルサイズ	ファイルを右クリック → プロパティ（または情報を見る）	30MB 以下
ページ数	PDF を開いてページ数を確認	100 ページ以下
パスワード保護	PDF を開いたときにパスワードを求められるか	パスワードなしが条件
ファイル形式	拡張子が .pdf になっているか	.pdf のみ対応
ネットワーク・ブラウザ	エラーが繰り返し起きるか	ページ再読み込みやブラウザ変更を試す



注釈:

### ページ数の上限について

Claude に一度に送れる PDF は **100 ページまで** です。100 ページを超えるファイルは、必要な章だけ切り出して送りましょう。

## 37.2 ケース別の対処法

### 37.2.1 ケース 1：ファイルサイズが大きすぎる（30MB 超）

**問題の症状：** アップロード時に「ファイルが大きすぎます」というエラーが出る、またはアップロードが完了しない。

**原因：** Claude に送れる 1 リクエストあたりの合計ファイルサイズは **30MB** が上限です。高解像度の画像を多く含む PDF や、スキャン品質が高い文書は、ページ数が少なくてもこの上限を超えることがあります。

**解決策：**

まずファイルサイズを確認します。Mac では Finder でファイルをクリックして「情報を見る (Command+I)」、Windows ではエクスプローラーで右クリック → 「プロパティ」で確認できます。

30MB を超えている場合は、次の方法で対処します。

#### 方法 1：PDF を圧縮する

無料で使える PDF 圧縮ツールを利用します。

- [ilovepdf.com](https://ilovepdf.com) (Web ブラウザで使える、インストール不要)
- [Smallpdf.com](https://smallpdf.com) (同様に Web ブラウザで利用可能)

いずれも「ファイルをアップロード → 圧縮 → ダウンロード」の 3 ステップです。

#### 方法 2：PDF を分割して必要な部分だけ送る

大きな資料の場合、関係する章やページだけを抜き出して送ると、サイズと処理精度の両方が改善します。分割手順は後述の「ページ数が多すぎる場合」を参照してください。

#### つまづきポイント：圧縮後も 30MB を超える場合

高解像度スキャンの PDF は、圧縮してもサイズが大きいたることがあります。その場合は PDF を章ごとに分割して複数回に分けて送るか、PDF の内容をテキストとしてコピーして貼り付ける方法を試してください。

### 37.2.2 ケース 2：ページ数が多すぎる（100 ページ超）

**問題の症状：** アップロードはできるが「ファイルのページ数が制限を超えています」というメッセージが出る、または途中のページ以降が読み取られない。

**原因：** Claude の 1 リクエストあたりのページ数上限は、多くの場合 **100 ページ** です。これを超えると後半ページが処理されません。

**解決策：必要な部分だけ抜き出す**

報告書や論文全体を送る必要はない場合がほとんどです。「エグゼクティブサマリー」「第 2 章だけ」のように、目的の部分だけを別ファイルとして保存して送りましょう。

PDF の特定ページを別ファイルとして保存する手順を OS ごとに説明します。

**Mac**

**Mac：プレビューアプリで特定ページを保存する**

1. PDF をプレビューアプリ（標準アプリ）で開きます
2. 左側のサムネイル一覧から、保存したいページをクリックして選択します
  - 連続した複数ページを選択するには、最初のページをクリックし、最後のページを Shift を押しながらクリック
  - バラバラのページは Command を押しながらクリック
3. 選択したサムネイルを右クリック → 「新規 PDF として書き出す」 を選択します
4. 保存先とファイル名を指定して保存します

これで選択したページだけが入った新しい PDF ファイルが作成されます。

## Windows

### Windows：ブラウザの印刷機能で特定ページを PDF として保存する

1. PDF ファイルをブラウザ（Microsoft Edge または Google Chrome）でドラッグ&ドロップして開きます
2. キーボードの **Ctrl + P**（印刷）を押します
3. 印刷ダイアログが開いたら、以下の設定をします：
  - **プリンター**：「Microsoft Print to PDF」（または「PDF として保存」）を選択
  - **ページ**：「カスタム」を選び、保存したいページ番号を入力（例：5-20 または 1,3,5）
4. 「印刷」または「保存」をクリックして、保存先を指定します

印刷先を「PDF」に設定することで、指定したページだけを新しい PDF として保存できます。



注釈:

#### 章ごとに分割して送る場合のコツ

同じ資料を複数回に分けて送るときは、「先ほど送った第 1 章に続いて、第 2 章についての同じ質問をします」のように前の会話を参照すると、一貫した回答が得られます。または、[Projects 機能](#)（ページ 121）を使うと、複数のファイルを登録して横断的に質問できます。

## 37.2.3 ケース 3：パスワード保護付き PDF が読めない

**問題の症状：**アップロードしても「このファイルは読み取れません」「暗号化されています」というエラーが出る。

**原因：**セキュリティのためにパスワードがかかっている PDF は、Claude には読み取れません。Claude に渡す前に、パスワードを解除したコピーを作成する必要があります。

**解決策：**パスワードを解除して保存し直す



警告:

パスワードを解除できるのは、あなた自身がパスワードを知っている場合に限りです。第三者が設定したパスワードを勝手に解除することは、セキュリティポリシーや法令に違反する場合があります。

## Mac

### Mac：プレビューアプリでパスワードを解除する

1. PDF をダブルクリックしてプレビューアプリで開きます
2. パスワードの入力を求められたら、正しいパスワードを入力して開きます
3. メニューバーから「ファイル」→「書き出す...」を選択します
4. 保存ダイアログが開きます。「暗号化」のチェックボックスがオフになっていることを確認します（チェックが入っていたら外してください）
5. 任意のファイル名を付けて「保存」をクリックします

保存された新しいファイルにはパスワードがかかっていません。このファイルを Claude に送ります。

## Windows

### Windows：ブラウザの印刷機能でパスワードを解除する

1. PDF ファイルをブラウザ（Microsoft Edge または Google Chrome）でドラッグ&ドロップして開きます
2. パスワードの入力を求められたら、正しいパスワードを入力して開きます
3. キーボードの **Ctrl + P**（印刷）を押します
4. 印刷ダイアログが開いたら、プリンターを「Microsoft Print to PDF」または「PDF として保存」に変更します
5. 「印刷」または「保存」をクリックして、保存先を指定します

ブラウザが PDF の内容を描画し直して出力するため、パスワードが取り除かれた新しい PDF ファイルが作成されます。

## 37.2.4 ケース 4：スキャン PDF で文字が正しく読み取れない

**問題の症状：** Claude が「文字が読み取れません」と言う、または内容が明らかに間違っている・欠けている。

**原因：** コピー機やスキャナーで作成した PDF は、文字データではなく**画像の集まり**です（「スキャン PDF」と呼びます）。Claude はページを画像として解析できるため、ある程度の読み取りは可能ですが、手書き文字・かすれたフォント・斜めのスキャンなどは精度が下がります。

### 解決策 1：プロンプトでカバーする

完全な読み取りを期待するのではなく、読み取り制限を前提にした質問の仕方をするとう�효です。

このスキャン文書から読み取れる範囲で、以下の情報を抽出してください。  
読み取りが不明確な箇所は「[不明]」と記載してください。

- 文書の作成日
- 宛先・差出人
- 本文の要旨（3 行以内）

### 解決策 2：OCR（文字認識）ソフトで変換してから送る

OCR（光学文字認識）とは、画像の中の文字をコンピューターが読めるテキストデータに変換する技術のことです。OCR 処理済みの PDF は文字情報が含まれるため、Claude の読み取り精度が大幅に上がります。

無料で使える OCR ツールの例：

- **Adobe Acrobat Reader**（無料版でも OCR 機能あり）：ツール → スキャンと OCR
- **Google Drive**：PDF をアップロードし、右クリック → 「Google ドキュメントとして開く」 → Google ドキュメントとして開いてから PDF で再保存
- **ilovepdf.com**：OCR 機能があります（月に一定回数まで無料）

#### つまづきポイント：OCR 後も精度が低い

手書き文字や非常に低解像度のスキャンは、OCR 処理後も誤変換が多く残ることがあります。その場合は、重要な数値や固有名詞を Claude が回答した後に確認する習慣をつけると安全です。「この数字は原文の何ページに記載されていますか？」のように出典を確認する質問も有効です。

### 37.2.5 ケース 5：「ファイルのダウンロードに失敗しました」エラー

**問題の症状：** アップロード中に「ファイルのダウンロードに失敗しました」「アップロードエラー」などのメッセージが表示される。

**原因：** ほとんどの場合、ブラウザや一時的な通信の問題が原因です。ファイル自体には問題がないことが多いです。

#### 解決策：

1. **ページを再読み込みして再試行する**（最も効果的）ブラウザで F5 キー（Mac は Command+R）を押してページを再読み込みし、もう一度ファイルをアップロードします。
2. **別のブラウザで試す** Chrome、Edge、Firefox などに切り替えると解消することがあります。
3. **ネットワークを確認する** Wi-Fi の接続が不安定な場合は、有線 LAN に切り替えるか、一度 Wi-Fi をオフにして再接続してみましょう。
4. **ブラウザのキャッシュをクリアする** ブラウザの設定から「閲覧データの削除」→「キャッシュされた画像とファイル」をクリアした後に試します。



注釈:

#### 同じエラーが続く場合

何度試しても同じエラーが出る場合は、ファイル自体に問題がある可能性があります。PDF ファイルをいったん別の PDF 変換ツール（Word → PDF の変換など）で作り直してから送ってみてください。

### 37.2.6 ケース 6：文字化け・レイアウトの崩れ

**問題の症状：** Claude の回答に意味不明な文字列が混じっている、または表やリストの構造が崩れて解釈されている。

**原因：**

- **文字化け：** 特殊なフォントや文字コードを使った PDF、古い日本語フォントを使った文書で起きやすいです。PDF を作成したソフトによっては、文字が正しくエクスポートされないことがあります。
- **レイアウト崩れ：** 複数列のレイアウト、テキストボックスが複雑に組み合わさった文書、スライドを PDF 化したものなどは、読み取り順序が崩れることがあります。

**解決策：**

#### 文字化けの場合

PDF をいったん Word や Google ドキュメントで開き（ファイル → 開く → PDF を選択）、そこからテキストをコピーして Claude に貼り付ける方法が確実です。テキストとして貼り付けることで、フォントの問題を回避できます。

#### レイアウト崩れの場合

複雑なレイアウトの文書は、Claude に読み取らせる際に手がかりを与えると精度が改善します。

この資料は 2 段組みのレイアウトです。  
左列から右列の順に読んでください。  
テーブルはそのままの行列構造を維持して内容を説明してください。

それでも難しい場合は、問題のある特定セクションだけをスクリーンショット（画像）として貼り付けて質問する方法もあります。

## 37.3 それでも解決しないときは

上記の方法を試しても問題が解決しない場合は、次の選択肢を試してみてください。

### 37.3.1 代替手段：テキストをコピーして貼り付ける

PDF をブラウザや PDF リーダーで開き、必要な部分のテキストをコピーして、Claude のチャット入力欄に直接貼り付ける方法が最も確実です。ファイルの添付よりも処理が安定します。

（テキストをここに貼り付け）

上記の文章を要約してください。

### 37.3.2 代替手段：スクリーンショットで送る

ページ数が少ない場合や、特定のグラフ・表だけを確認したい場合は、スクリーンショット（PNG または JPG 画像）として送る方法も有効です。Claude は画像も解析できます。

### 37.3.3 Anthropic サポートへの問い合わせ

上記をすべて試しても解決しない場合は、Anthropic のサポートに問い合わせることができます。

- サポートページ: <https://support.anthropic.com/>

問い合わせ時は以下の情報を伝えると、スムーズに対応してもらいやすくなります。

- 使用しているブラウザとバージョン (Chrome 120 など)
- エラーメッセージの全文 (スクリーンショットがあると尚良い)
- PDF のファイルサイズとページ数
- 発生した日時

## 37.4 次のステップ

- *Claude* で PDF を要約する方法 (ページ 315) — 問題が解決したら、要約テクニックを試してみましょう
- *Claude* で PDF を翻訳する方法 (ページ 337) — 英語の資料を日本語に翻訳する実践ガイド
- *Claude* で PDF・ファイルを活用する (ページ 309) — PDF 活用の基本と全体像に戻る





## 第38章 Claude でファイルを出力・生成する方法

「Claude に書いてもらった報告書、このまま上司に送りたい」「整理してもらったデータ、Excel で開きたい」——会話の中でできあがった文書やデータを、そのまま使える形で手元に保存したいと思ったことはありませんか。

Claude は PDF を「読む」だけでなく、文書やデータを「作って出力する」こともできます。この記事では、Claude が生成したコンテンツをさまざまなファイル形式で保存・活用する方法を解説します。

### 38.1 Artifacts（アーティファクト）とは

Claude が文書やコードを生成するとき、その結果がチャット画面の右側に専用のパネルで表示されます。この専用パネルで表示されるコンテンツを「Artifacts（アーティファクト）」と呼びます。

通常の会話と違い、Artifacts は：

- **プレビューで確認できる** — HTML や表であれば、実際に見た目を確認しながら中身を確認されます
- **ダウンロードできる** — パネル右下のダウンロードボタンから、ファイルとして保存できます
- **再利用できる** — 同じ会話の中で「もう少し修正して」と追加指示すると、Artifacts を更新してもらえます



注釈:

**Artifacts はデフォルトでオンになっています**

2024 年末以降の claude.ai では、Artifacts 機能はデフォルトで有効になっています。もし表示されない場合は、設定の「Capabilities（機能）」から「Artifacts」がオンになっているか確認してください。

### 38.2 Claude が出力できる形式

Claude（Artifacts 機能経由）で生成・ダウンロードできる主なファイル形式は以下のとおりです。

形式	拡張子	主な用途
HTML	.html	レポート・提案書・資料のひな型。ブラウザで開いて PDF に印刷できる
CSV	.csv	データ・集計結果。Excel や Google スプレッドシートで開ける

次のページに続く

表 1 – 前のページからの続き

形式	拡張子	主な用途
SVG	.svg	図・グラフ・ロゴなどのベクター画像。拡大しても綺麗なまま
Markdown	.md	軽量の文書形式。Notion や GitHub などにそのまま貼り付けられる
テキスト	.txt	プレーンテキスト。どのアプリでも開けるシンプルな形式
コード	.py / .js 等	プログラムのスクリプト。エンジニアとの共同作業に



注釈:

**Word・Excel・PowerPoint・PDF も直接生成できます（有料プランの一部）**

Claude の「ファイル作成機能」（コード実行が必要な機能）を有効にすると、.docx（Word）・.pptx（PowerPoint）・.xlsx（Excel）・PDF を直接ダウンロードできます。この機能は **Pro・Max・Team・Enterprise プラン** で利用可能です（設定 → Capabilities → 「コード実行とファイル作成」をオン）。無料プランをお使いの場合は、HTML や CSV で出力してから変換する方法が使えます（**無料プランでの代替手段**（ページ 335）を参照）。

### 38.3 実用例 1：レポート・提案書を PDF として保存する

「A4 で提出できる報告書を作りたい」という場面では、HTML で生成してブラウザから PDF として印刷する方法が便利です。

**ステップ 1：HTML 形式で作るよう依頼する**

以下の内容をもとに、A4 サイズで印刷できる HTML 形式の報告書を作成してください。  
見出し・箇条書き・表を使って、読みやすく整えてください。

- 【内容】
- ・プロジェクト名：〇〇システム導入
  - ・期間：2026 年 1 月～3 月
  - ・成果：コスト削減 20%、作業時間 30%短縮
  - ・課題：ユーザー教育に想定より時間がかかった

**ステップ 2：Artifacts のダウンロードボタンでファイルを保存する**

生成された HTML が Artifacts パネルに表示されたら、右下のダウンロードボタン（↓マーク）をクリックして .html ファイルを保存します。

**ステップ 3：ブラウザで開いて PDF に変換する**

保存した .html ファイルをダブルクリックしてブラウザで開き、Ctrl + P（Mac は Cmd + P）で印刷ダイアログを開きます。「送信先」を「PDF として保存」に変更して保存すれば完成です。

**つまづきポイント：印刷すると文字が小さすぎる・レイアウトが崩れる**

プロンプトに「A4 縦向きで印刷したときに読みやすいフォントサイズと余白を設定してください」と加えると改善されます。崩れが続く場合は「もう少し余白を増やして、フォントサイズを 14px にして」と具体的に修正依頼しましょう。

## 38.4 実用例 2：データを CSV で出力して Excel で開く

数字や一覧をまとめた結果を、Excel で活用したいときに便利です。

プロンプト例：

以下のデータを CSV 形式の Artifacts として出力してください。  
1 行目はヘッダー（商品名、単価、数量、合計）とし、  
数値はカンマを含まないようにしてください。

- ・ノート PC：98,000 円 × 5 台
- ・マウス：3,500 円 × 10 個
- ・USB ハブ：2,800 円 × 8 個

生成された CSV ファイルをダウンロードしたら、Excel で開くだけで表として使えます。



警告：

### Excel で文字化けしたときの対処法

CSV ファイルを Excel で直接ダブルクリックして開くと、日本語が文字化けすることがあります。この場合は、Excel の「データ」タブ → 「テキストまたは CSV から」を選び、文字コードに「UTF-8」を指定してインポートしてください。

## 38.5 実用例 3：SVG でグラフや図を画像として保存する

フローチャートや概念図、シンプルなグラフを「画像として使いたい」という場面では、SVG 形式が便利です。拡大しても画質が劣化しない点が強みです。

プロンプト例：

以下の業務フローを SVG 形式の図で表現してください。  
見やすい色と矢印を使い、各ステップを四角形で囲んでください。

1. 申請受付 → 2. 内容確認 → 3. 承認 → 4. 通知 → 5. 完了

ダウンロードした .svg ファイルは、PowerPoint や Word にそのまま貼り付けられます。また、ブラウザで開いて右クリック → 「名前を付けて画像を保存」で PNG や JPG に変換することもできます。

## 38.6 実用例 4：Word・PowerPoint を直接生成する（有料プラン）

Pro・Max・Team・Enterprise プランで「コード実行とファイル作成」が有効になっている場合、Word や PowerPoint を直接ダウンロードできます。

プロンプト例（Word 文書）：

以下の内容で Word 文書（.docx）を作成してダウンロードできるようにしてください。

- ・タイトル：2026 年度第 1 四半期 営業報告
- ・見出しは 2 段階（H1、H2）を使う

(次のページに続く)

(前のページからの続き)

- ・表を 1 つ含める（部門別の売上実績）
- ・フォントはデフォルトの日本語フォントを使用

### プロンプト例 (PowerPoint) :

以下の構成で PowerPoint ファイル (.pptx) を作ってください。

- ・スライド 1: タイトル「AI 活用推進計画」
- ・スライド 2: 目的と背景（箇条書き 3 点）
- ・スライド 3: 推進スケジュール（表形式）
- ・スライド 4: 期待される効果
- ・スライド 5: まとめ

シンプルで読みやすいデザインにしてください。



警告:

#### ファイル作成機能が使えない場合

「コード実行とファイル作成」がオフになっている、または無料プランを使用している場合、上記のプロンプトでは直接ファイルが生成されません。後述の代替手段をご利用ください。

## 38.7 きれいな PDF を作るプロンプトのコツ

HTML 経由で PDF を作る際、出来上がりのクオリティはプロンプトの書き方で大きく変わります。

### 効果的な指定の例 :

A4 縦向き、印刷時に余白 20mm で読みやすくなるよう、以下の条件で HTML を作成してください。

- ・フォント：本文 14px、見出し H2 は 18px、H3 は 16px
- ・行間：1.6
- ・表には薄いグレーのボーダーと交互の背景色をつける
- ・ページをまたぐとき、表や箇条書きの途中で切れないようにする
- ・印刷時に URL やナビゲーションが表示されないよう @media print で設定する

### 読者のイメージを伝えると仕上がりが変わります :

この提案書は、IT 知識のない経営層に読んでもらいます。  
専門用語は使わず、図解や箇条書きを多用した見やすいデザインにしてください。

## 38.8 無料プランでの代替手段

無料プランや、ファイル作成機能が使えない環境では、以下の方法で同様の成果が得られます。

### Word・PowerPoint が必要な場合：

Claude に Markdown または HTML で文書を作ってもらい、以下の方法で変換します。

1. **Markdown → Word**：「Pandoc」というフリーツールで変換できます。エンジニアでなくても使えるオンライン版（[markdowntodocx.com](https://markdowntodocx.com) など）があります
2. **HTML → Word**：Word で HTML ファイルを開くと、ある程度レイアウトを保ったまま .docx として保存できます
3. **PowerPoint の代替**：Claude に「各スライドの内容を箇条書きで書いて」と依頼し、PowerPoint のアウトラインビューに貼り付けると素早くスライドが作れます

### Excel が必要な場合：

CSV でダウンロードし、Excel で開いて整形するのが最もシンプルな方法です。

## 38.9 Artifacts が表示されない・ダウンロードできないときは

### つまづきポイント 1：Artifacts パネルが出てこない

claude.ai の設定（画面右上のアイコン → Settings → Capabilities）を開き、「Enable Artifacts」がオンになっているか確認してください。オフになっていると、生成物がチャットのテキストとして表示されるだけで、パネルや保存機能が使えません。

### つまづきポイント 2：ダウンロードボタンが見当たらない

Artifacts パネルの右下にある小さなアイコン群の中にダウンロードボタン（↓マーク）があります。パネルにカーソルを乗せると表示されることがあるため、マウスをパネル上で動かして確認してみてください。

### つまづきポイント 3：「Artifacts として出力して」と頼んでも普通のテキストで返ってくる

Claude は内容によって Artifacts で表示するか判断することがあります。「HTML 形式で」「CSV 形式の Artifacts として」など、**形式を明示した指定**をすると確実に Artifacts パネルに表示されます。

## 38.10 次のステップ

- *Claude* で PDF を要約する方法（ページ 315）— 要約した結果を PDF にまとめて保存するワークフローに
- *Claude* で PDF・ファイルを活用する（ページ 309）— ファイルを「渡す」基本に戻って確認する
- *Claude Web* の使い方（ページ 89）— Artifacts を含む Web 版の全機能を確認する



## 第39章 Claude で PDF を翻訳する方法

海外取引先からの提案書、英語で書かれた業界レポート、調達先の技術仕様書——英語の資料を読まなければ仕事が進まない場面は、職種を問わず増えています。

Google 翻訳や DeepL は手軽で便利なツールですが、専門用語が多い文書や、前後の文脈をつかまないと意味が取れない文章では、どうしても不自然な訳が出ることがあります。Claude は文書全体の内容を把握したうえで翻訳するため、専門分野の資料を「実務で使える日本語」にすることが得意です。

この記事では、PDF を Claude に渡して翻訳する手順と、分野ごとに使えるプロンプトテンプレートを紹介します。

### 39.1 基本の翻訳手順

まずは、PDF を添付して翻訳を依頼する最もシンプルな流れを確認しましょう。

#### ステップ 1: ファイルを添付する

claude.ai を開き、入力欄左のクリップアイコン (+ボタン) をクリックします。「Upload from computer (コンピューターからアップロード)」を選び、翻訳したい PDF を選択します。

#### ステップ 2: 翻訳の指示を入力して送信する

ファイルがアップロードされたら、入力欄に翻訳の指示を入力します。最低限、これだけでも翻訳が始まります。

添付の PDF を日本語に翻訳してください。



注釈:

#### 「翻訳してください」だけで試してみよう

まずはシンプルな指示で翻訳を試してみましょう。出力を見てから「もっとフォーマルに」「専門用語は英語のままにして」などの追加指示を出すことができます。一度のやりとりで完璧を目指さなくて大丈夫です。

### 39.2 分野別プロンプトテンプレート

翻訳の用途に合わせて指示を工夫すると、より実用的な翻訳が得られます。4つの分野別テンプレートを紹介します。

### 39.2.1 ビジネス文書（メール・報告書・提案書）

海外取引先からのメール、パートナーが作成した提案書、英語で書かれた会議資料などに使えます。ビジネスの場で読んで違和感のない日本語に仕上げることを優先します。

添付の英語ビジネス文書を日本語に翻訳してください。

- 文体：ビジネス文書として自然な日本語（「～します」「～ございます」調）
- 固有名詞（会社名、製品名）はそのまま残してください
- 翻訳が難しい表現やイディオムは、意識した上で原文を括弧内に添えてください

翻訳後、文書の主な内容を 3 点で箇条書きにまとめてください。



注釈:

#### 箇条書きのまとめを一緒に依頼するのがおすすめ

翻訳と同時に要点の整理を頼むことで、長い文書でも全体像をすぐ把握できます。上司への共有時や、自分がどこを精読すべきか判断するのにも役立ちます。

### 39.2.2 学術論文・研究レポート

医学論文、業界調査レポート、海外の学術誌——専門用語の密度が高く、直訳すると意味が通じにくい文書に使えます。

添付の英語論文を日本語に翻訳してください。

- 専門用語の扱い：初出時は「日本語訳（英語原文）」の形式で表記してください  
例：「機械学習（machine learning）」
- 略語が使われている場合は、初出時に正式名称を添えてください
- 著者名・雑誌名・固有名詞はそのまま残してください
- 翻訳後、以下の情報を整理してください：
  1. 研究の目的（2～3 文）
  2. 主な発見・結論（3～5 点）
  3. 今後の課題や限界として述べられていること

専門用語を英語と一緒に表記することで、社内での共有時も元の用語が引き継がれ、検索やリファレンスに便利です。

### 39.2.3 契約書・法的文書

海外取引先との契約書、利用規約、NDA（秘密保持契約）——法律文書は言葉の正確性が特に重要です。

添付の英語契約書を日本語に翻訳してください。

- 法的な意味が変わらないよう、できるだけ原文に忠実に訳してください
- 「shall」「shall not」「may」などの助動詞は、法的ニュアンスを保って訳してください

(次のページに続く)



(前のページからの続き)

(例: shall → 「～しなければならない」、may → 「～することができる」)

- 翻訳が一意に決まらない表現は、複数の解釈を示してください
- 翻訳後、特に注意が必要と思われる条項（義務、禁止事項、期限、違約金など）をリストアップしてください



危険:

**法的文書の翻訳は、必ず専門家に確認してもらってください**

契約書や法的文書の翻訳は、Claude を「下訳を作るツール」として使うのが安全です。内容の理解や論点の把握には役立ちますが、実際の契約に使う翻訳は必ず弁護士や翻訳の専門家にチェックしてもらいましょう。翻訳の誤りが重大なビジネスリスクにつながる可能性があります。

**39.2.4 技術仕様書・マニュアル**

ソフトウェアの仕様書、機器のマニュアル、API ドキュメント——IT や工学の専門用語が多い文書に使えます。

添付の英語技術文書を日本語に翻訳してください。

- IT・技術用語（API、SDK、クラウド、サーバーなど）はカタカナ表記を使ってください
- 手順を説明している箇所は、箇条書きや番号付きリストの構造をそのまま保持してください
- コードサンプルやコマンド例は翻訳せず、そのまま残してください
- 機能の名称（ボタン名、メニュー名など）は英語のまま括弧つきで残してください  
例: 「設定 (Settings) 画面を開き、保存 (Save) ボタンをクリックします」
- 翻訳が難しい技術用語があれば、その旨を注記してください

コード例やコマンドを翻訳してしまうと動作しなくなるため、「翻訳せずそのまま残す」と明示することがポイントです。

**39.3 翻訳精度を上げるテクニック**

基本の翻訳からさらに品質を高めたいときに使えるテクニックを紹介します。

**39.3.1 専門用語リストを一緒に渡す**

特定の訳語に統一したい専門用語がある場合は、用語集を一緒に渡します。これにより、文書全体で一貫した訳語が使われます。

添付の PDF を翻訳してください。以下の用語は必ず指定の訳語を使ってください。

用語リスト:

- liability → 賠償責任
- indemnification → 補償
- confidential information → 機密情報
- termination → 契約解除

(次のページに続く)

(前のページからの続き)

リスト以外の専門用語は文脈に合わせて適切に翻訳してください。

社内で使っている用語集がある場合は、その内容をテキストでそのまま貼り付けることもできます。

### 39.3.2 翻訳のトーンを指定する

同じ内容でも、読み手や使用目的に合わせてトーンを変えることができます。

添付の文書を日本語に翻訳してください。

【使用目的】 社内の一般社員向けに配布するため、専門知識がない読者にもわかりやすい平易な表現を使ってください。カタカナ語は避け、日本語で説明を加えながら訳してください。

添付の文書を日本語に翻訳してください。

【使用目的】 経営会議での報告資料。簡潔で格調ある文体を使い、余分な説明は省いて原文のニュアンスを忠実に保ってください。

### 39.3.3 難しい箇所に原文を併記させる

翻訳が難しい表現や、ニュアンスが失われる恐れがある部分を明示させることで、後で確認・修正しやすくなります。

翻訳が難しい箇所、意味が複数解釈できる箇所、または文化的背景が必要な表現があれば、翻訳後の日本語の直後に原文を括弧で添えてください。

例：「本契約の解釈については (the interpretation of this agreement)、当事者間での誠実な協議を経るものとします。」

### 39.3.4 分量の目安

Claude が 1 回のやりとりで処理できる PDF には、100 ページという上限があります。翻訳の場合、出力するテキスト量が多くなるため、実際には **20～30 ページ程度を目安**に分割して送ることをおすすめします。

#### つまづきポイント：長い PDF の翻訳が途中で止まる

長い文書を一度に翻訳しようとする、Claude の応答が途中で切れることがあります。「続きを翻訳してください」と追加メッセージを送ることで続きを取得できますが、最初から章ごとに分けて渡す方がスムーズです。特定のセクションだけを翻訳させたい場合は「第 3 章 (pages 12-18) だけ翻訳してください」と指定する方法もあります。

## 39.4 Google 翻訳・DeepL との使い分け

Claude が万能というわけではありません。それぞれのツールに得意・不得意があります。

ツール	得意なこと	向かないこと
Google 翻訳	素早い翻訳、短文・日常表現、多言語対応	専門的な文脈、長文の一貫性
DeepL	高品質な自然な翻訳、欧州言語、短～中程度の文書	長文書の文脈維持、細かい翻訳スタイルの指定
Claude	専門文書・長文の文脈理解、翻訳スタイルのカスタマイズ、要約との組み合わせ、用語統一	非常に短い表現の即時翻訳、日本語以外への翻訳精度

Claude が特に力を発揮する場面：

- 前後の文脈が重要な専門文書（契約書、論文、仕様書）
- 「専門用語はカタカナで」「原文の構造を保って」など細かい指示をしたいとき
- 翻訳と同時に「要点まとめ」「注意事項のリスト化」などをしたいとき
- 用語を社内の表記に統一したいとき

DeepL や Google 翻訳が向いている場面：

- ウェブページのざっくりした内容把握
- メールの返信を素早く確認したいとき
- 数語～数行程度の短い翻訳



注釈:

### DeepL で下訳して Claude で磨く方法もある

DeepL で素早く翻訳してから、その結果を Claude に「もっと自然なビジネス日本語に整えて」と依頼するという組み合わせも使えます。目的に合わせて使い分けてみてください。

## 39.5 よくあるつまづきポイント

### 直訳すぎて意味がわかりにくい

「直訳ではなく自然な日本語を優先してください」と明示的に指示するか、「ビジネス日本語として自然に」「読者は日本語ネイティブです」のように想定読者を伝えましょう。また、一度翻訳を受け取ってから「もっと自然な日本語に言い換えてください」とフォローアップするのも効果的です。

### 文書の中で同じ用語が別の訳語になっている

長い文書を翻訳すると、“contract” が「契約」「契約書」「合意書」と混在するようなことがあります。先述の「専門用語リストを一緒に渡す」テクニックを使うか、翻訳後に「この文書で使われている主な用語の訳語を一覧にしてください」と依頼して確認しましょう。

### 一部のページが翻訳されていない

スキャン PDF（スキャンした紙をそのまま PDF 化したもの）は、画像の文字認識精度に限界があります。「読み取れなかった箇所があれば教えてください」と依頼しておく、どこが欠けているか確認できます。文字のかすれや傾きが激しい場合は、可能であれば元の資料を再スキャンすることを検討してください。

## 39.6 次のステップ

- *Claude* で PDF を活用する（入門）（ページ 309）— ファイルのアップロード方法や基本操作を確認する
- *Claude* で PDF を要約する（ページ 315）— 長文を素早く把握したいときのテクニック
- ファイルを出力・生成する方法（ページ 331）— 翻訳した結果を PDF や Word で保存する
- プロンプトの基本テクニック（ページ 239）— 指示をよりの確に伝えるコツを学ぶ

## 第IV部

# 上級テクニック



## 第40章 MCP ガイド — Claude に外部ツールをつなげる

最終確認日: 2026 年 3 月 24 日 / MCP SDK: Python 1.x, TypeScript 1.x / Claude Desktop 最新版

Claude は標準状態でも高度な会話能力を持っていますが、「社内のデータベースを参照する」「GitHub のリポジトリを操作する」「ファイルシステムを読み書きする」といった外部との連携は、そのままでは行えません。

**MCP (Model Context Protocol)** はこの問題を解決するオープン標準です。USB-C ポートがさまざまなデバイスを同じ規格で接続できるように、MCP は AI アプリケーションとあらゆる外部システムを標準化された方法でつなぎます。

このガイドでは、MCP の仕組みを理解したうえで、実際に Claude Desktop や Claude Code へ MCP サーバーを接続する手順を解説します。

### 40.1 MCP とは — アーキテクチャの全体像

#### 40.1.1 3つの役割を理解する

MCP のエコシステムは **ホスト / クライアント / サーバー** という3つの役割で構成されます。

役割	説明	例
ホスト	AI モデルを実行し、MCP クライアントを内包するアプリケーション	Claude Desktop, Claude Code
クライアント	ホスト内でサーバーとの通信を担うコンポーネント	ホストに内蔵 (1 サーバーにつき 1 クライアント)
サーバー	外部システムへのアクセスを提供する軽量プロセス	Filesystem サーバー、GitHub サーバー等

Claude Desktop (ホスト)

└── MCP クライアント A ←→ Filesystem サーバー (ローカルファイル)  
└── MCP クライアント B ←→ GitHub サーバー (GitHub API)  
└── MCP クライアント C ←→ Slack サーバー (Slack API)

ホストは複数の MCP サーバーを同時に利用できます。Claude Desktop でファイルを読みながら GitHub を操作する、といったことも1つの会話の中でシームレスに行えます。

### 40.1.2 MCP サーバーが提供できる 3 種類の機能

MCP サーバーはクライアントに対して次の 3 種類のリソースを公開できます。

種類	説明	例
Resources (リソース)	ファイルのように読み取れるデータ	ファイル内容, DB レコード, API レス ポンス
Tools (ツール)	LLM が呼び出せる関数 (ユーザー承認付き)	ファイル書き込み, メール送信, コード 実行
Prompts (プロンプト)	再利用可能なプロンプトテンプレート	レポート生成用テンプレート等

### 40.1.3 なぜ直接 API 呼び出しではなく MCP なのか

直接 API を呼び出す方法の問題点:

- ツールごとに異なる認証方式・エラー処理・データ形式への対応が必要
- LLM にツールの使い方を学習させるためのプロンプトエンジニアリングが複雑化
- ツールを追加するたびにアプリケーション側のコード変更が必要

MCP を使う利点:

- **標準化:** すべてのサーバーが同じプロトコルで通信するため、一度書いた MCP クライアントはすべてのサーバーと互換
- **エコシステム:** コミュニティが作った数百のサーバーをすぐに利用可能
- **セキュリティ:** ホストはサーバーの機能 (使えるツール、アクセスできるデータ) を明示的に制御できる

## 40.2 既成サーバーの利用 — すぐに使えるサーバー一覧

Anthropic とコミュニティが提供する MCP サーバーは [github.com/modelcontextprotocol/servers](https://github.com/modelcontextprotocol/servers) にまとめられています。

### 40.2.1 Anthropic 公式が提供するサーバー

サーバー名	機能	用途例
Filesystem	ローカルファイルの読み書き	コードレビュー、ドキュメント整理
GitHub	リポジトリ・Issue・PR の操作	コードレビュー補助、Issue 管理
GitLab	GitLab プロジェクト操作	CI/CD パイプライン確認
Google Drive	Drive ファイルの検索・読取	ドキュメント参照
Google Maps	地図・場所情報の取得	住所検索、ルート案内
Slack	メッセージ送受信・チャンネル操作	通知自動化

次のページに続く



表 3 – 前のページからの続き

サーバー名	機能	用途例
PostgreSQL	PostgreSQL DB への読取専用アクセス	データ分析
SQLite	SQLite DB の読み書き	ローカルデータ管理
Puppeteer	ブラウザ自動操作	Web スクレイピング、E2E テスト
Fetch	Web ページの取得・変換	公開 API の呼び出し
Memory	ナレッジグラフによる記憶の永続化	コンテキスト跨ぎの記憶
AWS KB Retrieval	AWS Knowledge Base からの情報取得	RAG 構築

### 40.2.2 コミュニティサーバーの探し方

公式の一覧以外にも、以下のリソースでサーバーを探せます。

- [modelcontextprotocol.io/servers](https://modelcontextprotocol.io/servers) — 公式カタログ
- [mcp.so](https://mcp.so) — コミュニティが運営するサーバーディレクトリ
- npm / PyPI — @modelcontextprotocol/server- で始まるパッケージ

**つまづきポイント：「どのサーバーを使えばいいかわからない」**

まず用途から逆算してみてください。「ファイルを操作したい」→ Filesystem、「GitHub を使いたい」→ GitHub サーバーと対応は明快です。見つからない場合は npm で `mcp server` と検索するか、後述の自作サーバー入門を参照してください。

## 40.3 Claude Desktop での設定

Claude Desktop の MCP 設定は `claude_desktop_config.json` という JSON ファイルで管理します。

### 40.3.1 設定ファイルの場所

Mac

```
~/Library/Application Support/Claude/claude_desktop_config.json
```

ターミナルで開く場合:

```
open ~/Library/Application\ Support/Claude/
```

Windows

```
%APPDATA%\Claude\claude_desktop_config.json
```

エクスプローラーのアドレスバーに `%APPDATA%\Claude\` と入力して開けます。

ファイルが存在しない場合は新規作成してください。

### 40.3.2 設定ファイルの基本構造

```
{
 "mcpServers": {
 "サーバー名": {
 "command": "実行コマンド",
 "args": ["引数 1", "引数 2"],
 "env": {
 "環境変数名": "値"
 }
 }
 }
}
```

mcpServers の中に、使いたいサーバーをキーとして並べます。キー名（例: "filesystem"）は Claude の会話内でサーバーを識別する名前になります。

### 40.3.3 例 1: Filesystem サーバー（ローカルファイル操作）

```
まず npx でサーバーをインストール確認（初回は自動ダウンロード）
npx -y @modelcontextprotocol/server-filesystem --help
```

claude\_desktop\_config.json:

```
{
 "mcpServers": {
 "filesystem": {
 "command": "npx",
 "args": [
 "-y",
 "@modelcontextprotocol/server-filesystem",
 "/Users/yourname/Documents",
 "/Users/yourname/Projects"
]
 }
 }
}
```

args の最後に指定するパスが、Claude がアクセスできるディレクトリになります。複数指定可能です。セキュリティ上、必要最小限のディレクトリのみを指定することを推奨します。

### 40.3.4 例 2: GitHub サーバー

```
{
 "mcpServers": {
 "github": {
```

(次のページに続く)

(前のページからの続き)

```

 "command": "npx",
 "args": ["-y", "@modelcontextprotocol/server-github"],
 "env": {
 "GITHUB_PERSONAL_ACCESS_TOKEN": "ghp_XXXXXXXXXXXXXXXXXXXX"
 }
 }
}
}

```

GitHub の Personal Access Token は [github.com/settings/tokens](https://github.com/settings/tokens) で発行します。必要なスコープは操作内容によって異なりますが、読み取りのみなら `repo:read` で十分です。

### 40.3.5 例 3: 複数サーバーを同時に設定

```

{
 "mcpServers": {
 "filesystem": {
 "command": "npx",
 "args": [
 "-y",
 "@modelcontextprotocol/server-filesystem",
 "/Users/yourname/Projects"
]
 },
 "github": {
 "command": "npx",
 "args": ["-y", "@modelcontextprotocol/server-github"],
 "env": {
 "GITHUB_PERSONAL_ACCESS_TOKEN": "ghp_XXXXXXXXXXXXXXXXXXXX"
 }
 },
 "fetch": {
 "command": "uvx",
 "args": ["mcp-server-fetch"]
 }
 }
}

```

設定を保存後、**Claude Desktop を完全に再起動**する必要があります。再起動後、チャット画面の入力欄にハンマーアイコン（絵文字）が表示されれば MCP サーバーが認識されています。

#### つまづきポイント: ハンマーアイコンが表示されない

1. JSON の構文エラーがないか確認（カンマの有無、括弧の対応など）
2. `command` のパス（`npx`, `uvx`, `node` 等）が正しいか確認。which `npx` でフルパスを調べて指定するとよい場合があります

3. Claude Desktop を Dock から完全に終了し、再起動する
4. macOS では ~/Library/Logs/Claude/ にログファイルがあり、エラー内容を確認できます

## 40.4 Claude Code での設定

Claude Code では、プロジェクトのルートディレクトリに `.mcp.json` ファイルを置くことでプロジェクト単位の MCP 設定が可能です。この設定をリポジトリにコミットすれば、チーム全員が同じ MCP 環境を共有できます。

### 40.4.1 .mcp.json の構造

`.mcp.json` の構造は Claude Desktop の `claude_desktop_config.json` と同じ形式です。

```
{
 "mcpServers": {
 "filesystem": {
 "command": "npx",
 "args": [
 "-y",
 "@modelcontextprotocol/server-filesystem",
 "."
]
 },
 "github": {
 "command": "npx",
 "args": ["-y", "@modelcontextprotocol/server-github"],
 "env": {
 "GITHUB_PERSONAL_ACCESS_TOKEN": "${GITHUB_TOKEN}"
 }
 }
 }
}
```

"." を指定すると、現在のプロジェクトルートを対象にできます。環境変数は `${ENV_VAR_NAME}` の形式で参照可能です。実際の値は `.env` ファイルや CI/CD の環境変数として管理し、`.mcp.json` 自体にシークレットを書かないようにしましょう。

### 40.4.2 claude mcp コマンドによる管理

Claude Code では `claude mcp` コマンドで MCP サーバーを管理できます。

```
登録済みの MCP サーバー一覧を表示
claude mcp list
```

(次のページに続く)

(前のページからの続き)

```
MCP サーバーを追加 (プロジェクトスコープ)
claude mcp add filesystem npx -- -y @modelcontextprotocol/server-filesystem .

MCP サーバーを追加 (グローバルスコープ)
claude mcp add --scope global github npx -- -y @modelcontextprotocol/server-github

サーバーの詳細を確認
claude mcp get filesystem

サーバーを削除
claude mcp remove filesystem
```

スコープについて:

スコープ	設定ファイルの場所	適用範囲
project (デフォルト)	.mcp.json (プロジェクトルート)	そのプロジェクトのみ
local	~/.config/claude/mcp.json 等	ローカルマシン全体
global	グローバル設定ファイル	全プロジェクト

40.4.3 Claude Code での動作確認

Claude Code セッション内で /mcp スラッシュコマンドを使うと、接続中の MCP サーバーの状態を確認できます。

```
/mcp
```

ツールが正しく読み込まれていれば、利用可能なツール一覧が表示されます。

つまづきポイント: .mcp.json を追加したのにツールが使われない

Claude Code は起動時に .mcp.json を読み込みます。ファイルを追加・変更した後は、Claude Code のセッションを /exit で終了し、再起動してください。/mcp コマンドで接続状態を確認できます。

40.5 自作サーバー入門

既成のサーバーで対応できない場合は、自分で MCP サーバーを作れます。MCP SDK は TypeScript と Python の両方で提供されています。

40.5.1 TypeScript で作る最小限のサーバー

前提条件: Node.js 18 以上

```
mkdir my-mcp-server
cd my-mcp-server
npm init -y
npm install @modelcontextprotocol/sdk zod
npm install --save-dev typescript @types/node
npx tsc --init
```

src/index.ts:

```
import { McpServer } from "@modelcontextprotocol/sdk/server/mcp.js";
import { StdioServerTransport } from "@modelcontextprotocol/sdk/server/stdio.js";
import { z } from "zod";

// サーバーのインスタンスを作成
const server = new McpServer({
 name: "my-server",
 version: "1.0.0",
});

// ツールを定義する
server.tool(
 "greet",
 "指定した名前に挨拶する",
 {
 name: z.string().describe("挨拶する相手の名前"),
 },
 async ({ name }) => {
 return {
 content: [
 {
 type: "text",
 text: `こんにちは、${name}さん！`,
 },
],
 };
 }
);

// サーバーを起動
const transport = new StdioServerTransport();
await server.connect(transport);
console.error("Server started"); // STDIO サーバーでは stderr に出力すること
```

**重要:** STDIO 通信を使うサーバーでは、`console.log()` は絶対に使わないでください。stdout に書き込むと JSON-RPC メッセージが壊れてサーバーが動作しなくなります。デバッグ出力は必ず `console.error()` を使って stderr に書き出してください。

package.json に build スクリプトを追加してビルドします。

```
npx tsc
node dist/index.js
```

## 40.5.2 Python で作る最小限のサーバー

前提条件: Python 3.10 以上、uv パッケージマネージャー

```
uv init my-mcp-server
cd my-mcp-server
uv add "mcp[cli]"
```

server.py:

```
from mcp.server.fastmcp import FastMCP

mcp = FastMCP("my-server")

@mcp.tool()
def greet(name: str) -> str:
 """指定した名前に挨拶する。

 Args:
 name: 挨拶する相手の名前
 """
 return f"こんにちは、{name}さん！"

if __name__ == "__main__":
 mcp.run(transport="stdio")
```

```
uv run server.py
```

## 40.5.3 自作サーバーを Claude Desktop に登録する

ビルド・動作確認ができれば、claude\_desktop\_config.json に追加します。

```
{
 "mcpServers": {
 "my-server": {
 "command": "node",
 "args": ["/絶対パス/my-mcp-server/dist/index.js"]
 }
 }
}
```

Python の場合:

```
{
 "mcpServers": {
 "my-server": {
 "command": "uv",
 "args": [
 "--directory",
 "/絶対パス/my-mcp-server",
 "run",
 "server.py"
]
 }
 }
}
```

パスは必ず**絶対パス**で指定してください。相対パスは動作しません。

## 40.6 デバッグとトラブルシューティング

### 40.6.1 Claude Desktop のログを確認する

Mac

```
リアルタイムでログを確認
tail -f ~/Library/Logs/Claude/mcp*.log
```

Windows

```
ログフォルダを開く
explorer "$env:APPDATA\Claude\logs"
```

### 40.6.2 MCP Inspector でサーバーを単体テスト

サーバーをデプロイする前に、MCP Inspector という開発ツールで動作確認できます。

```
npx @modelcontextprotocol/inspector node dist/index.js
```

ブラウザが開き、サーバーのツール・リソース・プロンプトを GUI で確認・テストできます。

### 40.6.3 よくあるエラーと解決策



エラー	原因	解決策
ハンマーアイコンが表示されない	JSON 構文エラー / サーバーの起動失敗	ログを確認、JSON をバリデート
command not found	npx / uv のパスが通っていない	command にフルパスを指定 (例: /usr/local/bin/npx)
ツールは表示されるが実行でエラー	サーバー側のロジックエラー	Inspector で単体テスト、stderr ログを確認
GITHUB_TOKEN が認識されない	環境変数が渡されていない	"env" フィールドに明示的に記載
サーバーがすぐ終了する	stdout へのログ書き込み (STDIO の場合)	console.log → console.error に変更

#### つまづきポイント: Python の uv コマンドが見つからない

uv はまだ広く普及していないツールです。command に uv と書いただけでは macOS 上で見つからないことがあります。which uv でフルパスを確認し (例: /Users/yourname/.cargo/bin/uv)、そのフルパスを command フィールドに指定してください。

## 40.7 セキュリティの考慮事項

MCP は Claude の能力を大きく拡張しますが、それはリスクも伴います。本番環境での利用や他人が作ったサーバーを使う際には、以下の点に注意してください。

### 40.7.1 アクセス範囲を最小限にする

Filesystem サーバーを設定する際は、Claude がアクセスできるディレクトリを必要最小限にしてください。

```
// 悪い例: ホームディレクトリ全体へのアクセス
"args": ["-y", "@modelcontextprotocol/server-filesystem", "/Users/yourname"]

// 良い例: 特定のプロジェクトディレクトリのみ
"args": ["-y", "@modelcontextprotocol/server-filesystem", "/Users/yourname/my-project/docs"]
```

### 40.7.2 API トークンの権限を絞る

GitHub トークン等は最小権限の原則に従い、必要なスコープのみを付与します。読み取り専用で十分な場合は書き込み権限を与えないでください。

### 40.7.3 信頼できるサーバーのみを使う

コミュニティが作った MCP サーバーを使う前に、以下を確認してください。

- ソースコードが公開されているか (npm / PyPI からインストールする場合も、元のリポジトリを確認)
- 実績のある開発者・組織が管理しているか
- 必要以上のデータにアクセスしようとしていないか

#### 40.7.4 環境変数にシークレットを書かない

`.mcp.json` はリポジトリにコミットされることが多いため、シークレット（API キー、トークン）を直接書かないようにします。

```
// 悪い例：トークンを直書き
"env": { "GITHUB_TOKEN": "ghp_abc123..." }

// 良い例：環境変数参照
"env": { "GITHUB_TOKEN": "${GITHUB_TOKEN}" }
```

実際のトークンは `.env` ファイル（`.gitignore` に追加）や CI/CD のシークレット管理で管理します。

#### 40.7.5 ツール実行の承認フロー

MCP のツール（Tools）は、Claude が実行しようとするとき Claude Desktop / Claude Code 上でユーザーに承認を求めるダイアログが表示されます（ホスト実装によります）。重要な操作（ファイル削除、外部サービスへの書き込み等）を自動承認しないよう、ホストの設定を確認してください。

### 40.8 次のステップ

MCP の基本を習得したら、以下のリソースで理解を深められます。

- [API 入門 — Claude をアプリに組み込む](#)（ページ 357） — Claude API を直接呼び出してアプリを構築する方法。MCP と組み合わせることでさらに強力なシステムを作れます
- [MCP 公式サーバーカタログ](#) — コミュニティが作った MCP サーバーの一覧
- [MCP アーキテクチャ詳細](#) — トランスポート層（STDIO vs HTTP/SSE）やプロトコルの詳細
- [MCP SDK リファレンス（TypeScript）](#) — TypeScript SDK の API ドキュメント
- [MCP SDK リファレンス（Python）](#) — Python SDK の API ドキュメント

# 第41章 API 入門 — Claude をアプリに組み込む

Anthropic の Claude API を使えば、自分のアプリケーションやサービスに Claude の AI 能力を組み込みます。チャット UI ではできない、バックエンド処理の自動化や独自インターフェースの構築、大量テキストの一括処理など、さまざまな可能性が広がります。

この記事では、API キーの取得から実際のリクエスト送信、Function Calling、ストリーミング、コスト最適化まで、実践的な内容を Python と TypeScript のコード例で解説します。

## 41.1 API の概要

### 41.1.1 エンドポイント体系

Claude API のベース URL は `https://api.anthropic.com/v1` です。主なエンドポイントは以下のとおりです。

エンドポイント	用途
POST /v1/messages	テキスト生成（最も基本的な API）
POST /v1/messages/batches	バッチ処理（大量リクエストを非同期処理）
POST /v1/messages/count_tokens	トークン数の事前カウント

### 41.1.2 API キーの取得

Anthropic Console にアクセスし、アカウントを作成します。「API Keys」セクションで新しいキーを生成できます。



警告:

**API キーは厳重に管理してください。** API キーが漏洩すると、第三者があなたのアカウントでリクエストを送り、費用が発生します。以下のルールを守ってください。

- API キーをソースコードにハードコーディングしない
- Git リポジトリ（特に公開リポジトリ）にコミットしない
- .env ファイルは .gitignore に追加する
- 本番環境では環境変数や Secret Manager（AWS Secrets Manager、Google Secret Manager 等）を使う

### 41.1.3 レート制限

Claude API にはリクエスト数 (RPM) とトークン数 (TPM) のレート制限があります。制限値はアカウントの利用状況や Tier によって異なります。制限に達した場合、API は **429 Too Many Requests** を返します。

**つまづきポイント:** 本番アプリで 429 エラーが頻発する場合は、指数バックオフ（リトライ間隔を徐々に延ばす）で再試行するロジックを組み込んでください。Anthropic の公式 SDK にはこの機能が組み込まれています。

## 41.2 クイックスタート

### 41.2.1 SDK のインストール

まず、Anthropic の公式 SDK をインストールします。

#### Mac

##### Python:

```
pip install anthropic
```

##### TypeScript / Node.js:

```
npm install @anthropic-ai/sdk
```

#### Windows

##### Python:

```
pip install anthropic
```

##### TypeScript / Node.js:

```
npm install @anthropic-ai/sdk
```

### 41.2.2 環境変数の設定

API キーを環境変数に設定します。

```
export ANTHROPIC_API_KEY="sk-ant-..."
```

本番アプリでは `.env` ファイルを使い、`.gitignore` に追加することを推奨します。

```
.env
ANTHROPIC_API_KEY=sk-ant-...
```

### 41.2.3 最初のリクエスト

Python:

```
import anthropic

client = anthropic.Anthropic()

message = client.messages.create(
 model="claude-opus-4-6",
 max_tokens=1024,
 messages=[
 {"role": "user", "content": "Hello, Claude!"}
]
)

print(message.content[0].text)
```

TypeScript:

```
import Anthropic from "@anthropic-ai/sdk";

const client = new Anthropic();

const message = await client.messages.create({
 model: "claude-opus-4-6",
 max_tokens: 1024,
 messages: [{ role: "user", content: "Hello, Claude!" }],
});

console.log(message.content[0].text);
```

cURL (API の動作確認に便利) :

```
curl https://api.anthropic.com/v1/messages \
-H "x-api-key: $ANTHROPIC_API_KEY" \
-H "anthropic-version: 2023-06-01" \
-H "content-type: application/json" \
-d '{
 "model": "claude-opus-4-6",
 "max_tokens": 1024,
 "messages": [
 {"role": "user", "content": "Hello, Claude!"}
]
}'
```

レスポンスは以下のような JSON 形式で返ってきます。

```
{
 "id": "msg_01XFDUDYJgAACzvnptvVoYEL",
 "type": "message",
 "role": "assistant",
 "content": [
 {
 "type": "text",
 "text": "Hello! How can I assist you today?"
 }
],
 "model": "claude-opus-4-6",
 "stop_reason": "end_turn",
 "usage": {
 "input_tokens": 10,
 "output_tokens": 12
 }
}
```

### 41.3 Messages API の基本

#### 41.3.1 リクエストパラメータ

/v1/messages エンドポイントの主要なパラメータです。

パラメータ	必須	説明
model	○	使用するモデル ID (例: claude-opus-4-6)
max_tokens	○	生成するトークンの最大数
messages	○	会話履歴の配列 (role と content のペア)
system	—	システムプロンプト (モデルの役割・制約を設定)
temperature	—	出力のランダム性 (0~1、デフォルト 1)
top_p	—	核サンプリングのしきい値
stop_sequences	—	生成を停止するシーケンスの配列
stream	—	ストリーミングを有効にするか (true/false)

#### 41.3.2 システムプロンプト

システムプロンプトを使うと、モデルに役割や制約を設定できます。

Python:

```
message = client.messages.create(
 model="claude-opus-4-6",
 max_tokens=1024,
 system="あなたは日本語で回答するカスタマーサポートエージェントです。丁寧かつ簡潔に回答し
```

(次のページに続く)

(前のページからの続き)

```
てください。",
 messages=[
 {"role": "user", "content": "返品ポリシーを教えてください。"}
]
)
```

**TypeScript:**

```
const message = await client.messages.create({
 model: "claude-opus-4-6",
 max_tokens: 1024,
 system:
 "あなたは日本語で回答するカスタマーサポートエージェントです。丁寧かつ簡潔に回答してください。",
 messages: [{ role: "user", content: "返品ポリシーを教えてください。" }],
});
```

### 41.3.3 マルチターン会話

会話の履歴を `messages` 配列に積み上げることでマルチターン会話を実現します。

**Python:**

```
conversation_history = []

def chat(user_message: str) -> str:
 conversation_history.append({
 "role": "user",
 "content": user_message
 })

 response = client.messages.create(
 model="claude-opus-4-6",
 max_tokens=1024,
 messages=conversation_history
)

 assistant_message = response.content[0].text
 conversation_history.append({
 "role": "assistant",
 "content": assistant_message
 })

 return assistant_message
```

(次のページに続く)

(前のページからの続き)

```
使用例
print(chat("自己紹介してください。"))
print(chat("あなたの得意なことは何ですか？"))
```

**TypeScript:**

```
const conversationHistory: { role: "user" | "assistant"; content: string }[] =
 [];

async function chat(userMessage: string): Promise<string> {
 conversationHistory.push({ role: "user", content: userMessage });

 const response = await client.messages.create({
 model: "claude-opus-4-6",
 max_tokens: 1024,
 messages: conversationHistory,
 });

 const assistantMessage = (response.content[0] as { text: string }).text;
 conversationHistory.push({ role: "assistant", content: assistantMessage });

 return assistantMessage;
}

// 使用例
console.log(await chat("自己紹介してください。"));
console.log(await chat("あなたの得意なことは何ですか？"));
```

**つまづきポイント:** 会話履歴を蓄積し続けると、いずれかのリクエストでコンテキストウィンドウの上限（モデルによって異なり、Claude Opus 4.6 / Sonnet 4.6 は最大 100 万トークン、Haiku 4.5 は 200K トークン）を超えます。長期的な会話では、古いメッセージを削除するか、要約してから新しいリクエストに渡す設計が必要です。

**41.3.4 temperature の使い分け**

temperature は出力の多様性を制御します。

temperature	向いている用途
0.0	分析、分類、事実確認（決定論的な出力が必要な場合）
0.3～0.7	チャットボット、Q&A（バランスの取れた出力）
0.8～1.0	創作、ブレインストーミング（多様な出力が欲しい場合）



## 41.4 Function Calling（ツール使用）

Function Calling（ツール使用）は、Claude に外部関数やツールを呼び出す能力を与える機能です。例えば、現在の天気情報の取得、データベースの検索、計算処理など、Claude 単体ではできない処理を組み込みます。

### 41.4.1 実行フロー

1. アプリが「ツール定義」を含むリクエストを送る
2. Claude がツールを呼び出す必要があると判断したら `tool_use` ブロックを返す
3. アプリが実際の関数を実行し、結果を Claude に返す
4. Claude が結果を使って最終的な回答を生成する

### 41.4.2 ツール定義の書き方

Python:

```
tools = [
 {
 "name": "get_weather",
 "description": "指定した都市の現在の天気情報を取得します。",
 "input_schema": {
 "type": "object",
 "properties": {
 "city": {
 "type": "string",
 "description": "天気を調べる都市名（例：東京、大阪）"
 },
 "unit": {
 "type": "string",
 "enum": ["celsius", "fahrenheit"],
 "description": "温度の単位"
 }
 },
 "required": ["city"]
 }
 }
]

response = client.messages.create(
 model="claude-opus-4-6",
 max_tokens=1024,
 tools=tools,
 messages=[
 {"role": "user", "content": "東京の今日の天気を教えてください。"}
]
)
```

(次のページに続く)

(前のページからの続き)

```

)

ツール呼び出しが返された場合の処理
if response.stop_reason == "tool_use":
 tool_use_block = next(
 block for block in response.content if block.type == "tool_use"
)
 tool_name = tool_use_block.name
 tool_input = tool_use_block.input
 tool_use_id = tool_use_block.id

実際のツール処理（ここでは仮のレスポンス）
if tool_name == "get_weather":
 tool_result = {
 "temperature": 22,
 "condition": "晴れ",
 "humidity": 60
 }

ツール実行結果を Claude に返す
final_response = client.messages.create(
 model="claude-opus-4-6",
 max_tokens=1024,
 tools=tools,
 messages=[
 {"role": "user", "content": "東京の今日の天気を教えてください。"},
 {"role": "assistant", "content": response.content},
 {
 "role": "user",
 "content": [
 {
 "type": "tool_result",
 "tool_use_id": tool_use_id,
 "content": str(tool_result)
 }
]
 }
]
)

print(final_response.content[0].text)

```

TypeScript:

```

const tools: Anthropic.Messages.Tool[] = [
 {

```

(次のページに続く)

(前のページからの続き)

```

name: "get_weather",
description: "指定した都市の現在の天気情報を取得します。",
input_schema: {
 type: "object",
 properties: {
 city: {
 type: "string",
 description: "天気を調べる都市名（例：東京、大阪）",
 },
 unit: {
 type: "string",
 enum: ["celsius", "fahrenheit"],
 description: "温度の単位",
 },
 },
 required: ["city"],
},
];

const response = await client.messages.create({
 model: "claude-opus-4-6",
 max_tokens: 1024,
 tools,
 messages: [{ role: "user", content: "東京の今日の天気を教えてください。" }],
});

if (response.stop_reason === "tool_use") {
 const toolUseBlock = response.content.find(
 (block): block is Anthropic.Messages.ToolUseBlock =>
 block.type === "tool_use"
)!;

 // 実際のツール処理（ここでは仮のレスポンス）
 const toolResult = { temperature: 22, condition: "晴れ", humidity: 60 };

 const finalResponse = await client.messages.create({
 model: "claude-opus-4-6",
 max_tokens: 1024,
 tools,
 messages: [
 { role: "user", content: "東京の今日の天気を教えてください。" },
 { role: "assistant", content: response.content },
 {
 role: "user",

```

(次のページに続く)

(前のページからの続き)

```

 content: [
 {
 type: "tool_result",
 tool_use_id: toolUseBlock.id,
 content: JSON.stringify(toolResult),
 },
],
 },
],
});
console.log(finalResponse.content[0]);
}

```

**つまづきポイント:** `stop_reason` が `"tool_use"` でも `"end_turn"` でもない場合 (例: `"max_tokens"`) は、`tool_use` ブロックがないままに生成が打ち切られています。`max_tokens` を増やすか、ツールの呼び出しが必要かを事前に判断するロジックを追加してください。

## 41.5 ストリーミング

ストリーミングを使うと、レスポンスが完全に生成される前にトークンを逐次受け取れます。チャットアプリや長文生成で、ユーザーが結果を待ち続けることなくリアルタイムに表示できます。

Claude API は SSE (Server-Sent Events) 形式でストリーミングをサポートしています。

**Python:**

```

with client.messages.stream(
 model="claude-opus-4-6",
 max_tokens=1024,
 messages=[
 {"role": "user", "content": "量子コンピュータについて詳しく説明してください。"}
]
) as stream:
 for text in stream.text_stream:
 print(text, end="", flush=True)
 print() # 最後に改行

ストリーム完了後に最終メッセージを取得
final_message = stream.get_final_message()
print(f"\n トークン使用量: {final_message.usage}")

```

**TypeScript:**

```

const stream = await client.messages.stream({
 model: "claude-opus-4-6",
 max_tokens: 1024,

```

(次のページに続く)

(前のページからの続き)

```

messages: [
 {
 role: "user",
 content: "量子コンピュータについて詳しく説明してください。",
 },
],
});

for await (const chunk of stream) {
 if (
 chunk.type === "content_block_delta" &&
 chunk.delta.type === "text_delta"
) {
 process.stdout.write(chunk.delta.text);
 }
}

// ストリーム完了後に最終メッセージを取得
const finalMessage = await stream.getFinalMessage();
console.log("\n トークン使用量:", finalMessage.usage);

```

### 41.5.1 Next.js での Server-Sent Events 実装例

TypeScript (Next.js App Router) :

```

// app/api/chat/route.ts
import Anthropic from "@anthropic-ai/sdk";

const client = new Anthropic();

export async function POST(req: Request) {
 const { messages } = await req.json();

 const encoder = new TextEncoder();

 const stream = new ReadableStream({
 async start(controller) {
 const anthropicStream = await client.messages.stream({
 model: "claude-opus-4-6",
 max_tokens: 1024,
 messages,
 });

 for await (const chunk of anthropicStream) {

```

(次のページに続く)

(前のページからの続き)

```

 if (
 chunk.type === "content_block_delta" &&
 chunk.delta.type === "text_delta"
) {
 controller.enqueue(encoder.encode(chunk.delta.text));
 }
}
controller.close();
},
});

return new Response(stream, {
 headers: { "Content-Type": "text/plain; charset=utf-8" },
});
}

```

## 41.6 Extended Thinking (拡張思考)

Extended Thinking は、Claude が最終的な回答を出す前に内部的に思考プロセスを展開する機能です。複雑な数学問題、多段階の推論、詳細な分析など、高精度が求められるタスクで特に効果を発揮します。

thinking パラメータに `budget_tokens` (思考に使えるトークンの最大数) を指定して有効化します。

**Python:**

```

response = client.messages.create(
 model="claude-opus-4-6",
 max_tokens=16000,
 thinking={
 "type": "enabled",
 "budget_tokens": 10000 # 思考プロセスに使えるトークン数
 },
 messages=[
 {
 "role": "user",
 "content": "次の数学の証明を行ってください: 素数が無限に存在することを証明してく
ださい。"
 }
]
)

レスポンスには thinking ブロックと text ブロックが含まれる
for block in response.content:
 if block.type == "thinking":
 print("=== 思考プロセス ===")

```

(次のページに続く)

(前のページからの続き)

```

 print(block.thinking)
elif block.type == "text":
 print("=== 最終回答 ===")
 print(block.text)

```

**TypeScript:**

```

const response = await client.messages.create({
 model: "claude-opus-4-6",
 max_tokens: 16000,
 thinking: {
 type: "enabled",
 budget_tokens: 10000,
 },
 messages: [
 {
 role: "user",
 content:
 "次の数学の証明を行ってください: 素数が無限に存在することを証明してください。",
 },
],
});

for (const block of response.content) {
 if (block.type === "thinking") {
 console.log("=== 思考プロセス ===");
 console.log(block.thinking);
 } else if (block.type === "text") {
 console.log("=== 最終回答 ===");
 console.log(block.text);
 }
}

```

**注釈:**

Extended Thinking は claude-opus-4-6 などの対応モデルでのみ利用できます。budget\_tokens を大きくするほど推論精度が上がりますが、その分コストも増加します。用途に応じて適切な値を設定してください。

## 41.7 バッチ処理

大量のリクエストを処理する場合、Message Batches API を使うと効率的です。通常の API と比べて**最大 50% のコスト削減**が可能で、非同期でバックグラウンド処理されます。

### 41.7.1 バッチリクエストの送信

**Python:**

```
バッチジョブの作成
batch = client.messages.batches.create(
 requests=[
 {
 "custom_id": "request-1",
 "params": {
 "model": "claude-opus-4-6",
 "max_tokens": 1024,
 "messages": [
 {"role": "user", "content": "東京について教えてください。"}
]
 }
 },
 {
 "custom_id": "request-2",
 "params": {
 "model": "claude-opus-4-6",
 "max_tokens": 1024,
 "messages": [
 {"role": "user", "content": "大阪について教えてください。"}
]
 }
 }
]
)

print(f"バッチ ID: {batch.id}")
print(f"ステータス: {batch.processing_status}")
```

**TypeScript:**

```
const batch = await client.messages.batches.create({
 requests: [
 {
 custom_id: "request-1",
 params: {
 model: "claude-opus-4-6",
 max_tokens: 1024,
 messages: [{ role: "user", content: "東京について教えてください。" }],
 },
 },
 {
 custom_id: "request-2",
```

(次のページに続く)



(前のページからの続き)

```

 params: {
 model: "claude-opus-4-6",
 max_tokens: 1024,
 messages: [{ role: "user", content: "大阪について教えてください。" }],
 },
],
});

console.log(`バッチ ID: ${batch.id}`);
console.log(`ステータス: ${batch.processing_status}`);

```

### 41.7.2 バッチ結果の取得

バッチが完了するまで数分～数時間かかる場合があります。ポーリングで状態を確認します。

Python:

```

import time

バッチの完了待ち（ポーリング）
while True:
 batch = client.messages.batches.retrieve(batch.id)
 if batch.processing_status == "ended":
 break
 print(f"処理中... ({batch.request_counts.processing} 件残り)")
 time.sleep(60) # 1分待つて再確認

結果の取得
for result in client.messages.batches.results(batch.id):
 print(f"ID: {result.custom_id}")
 if result.result.type == "succeeded":
 print(result.result.message.content[0].text)
 else:
 print(f"エラー: {result.result.error}")

```

**つまづきポイント:** バッチ処理は最大 24 時間以内に完了しますが、それを超えるとタイムアウトします。処理件数が多い場合は複数の小さなバッチに分割することを検討してください。1 バッチあたりの最大リクエスト数は 10,000 件です。

## 41.8 コスト最適化

### 41.8.1 料金体系（概要）

Claude の API 料金は入力トークン数と出力トークン数に応じた従量課金制です。以下は 2026 年 3 月時点の目安です。**最新の料金は必ず [Anthropic の公式料金ページ](#) で確認してください。**

モデル	入力 (100 万トークンあたり)	出力 (100 万トークンあたり)	特徴
Claude Haiku 4.5	\$1.00	\$5.00	最速・最低コスト。シンプルなタスク向け
Claude Sonnet 4.6	\$3.00	\$15.00	性能とコストのバランスが優れる
Claude Opus 4.6	\$5.00	\$25.00	最高性能。複雑な推論・分析向け



警告:

上記の料金は目安です。Anthropic は料金を変更することがあります。実際の料金は <https://www.anthropic.com/pricing> で確認してください。

41.8.2 Prompt Caching でコストを削減

Prompt Caching は、繰り返し使う長いシステムプロンプトやコンテキストをキャッシュする機能です。キャッシュヒット時は入力トークンのコストが大幅に削減されます (約 90% オフ)。

長いシステムプロンプト、大量のドキュメント参照、Few-shot 例の繰り返しなど、毎回同じ内容を送るケースで効果的です。

Python:

```
response = client.messages.create(
 model="claude-opus-4-6",
 max_tokens=1024,
 system=[
 {
 "type": "text",
 "text": "あなたは日本の法律に精通した専門家 AI アシスタントです。\\n\\n" + very_
↪long_legal_document, # 長い参照ドキュメント
 "cache_control": {"type": "ephemeral"} # キャッシュを有効化
 }
],
 messages=[
 {"role": "user", "content": "第 3 条の解釈を教えてください。"}
]
)

キャッシュの使用状況を確認
print(f"キャッシュ書き込みトークン: {response.usage.cache_creation_input_tokens}")
print(f"キャッシュ読み込みトークン: {response.usage.cache_read_input_tokens}")
```

### 41.8.3 モデル選定のガイドライン

ユースケース	推奨モデル	理由
分類・ルーティング・シンプルな Q&A	Claude Haiku 4.5	速度とコストを重視
チャットボット・文章生成・コード補完	Claude Sonnet 4.6	バランスが最も優れる
複雑な推論・分析・研究	Claude Opus 4.6	精度最優先
バッチ処理（大量の単純タスク）	Claude Haiku 4.5	低コストで大量処理

### 41.8.4 トークン数を事前に確認する

リクエストを送る前に、`/v1/messages/count_tokens` でトークン数を確認できます。

**Python:**

```
token_count = client.messages.count_tokens(
 model="claude-opus-4-6",
 messages=[
 {"role": "user", "content": "このリクエストは何トークンですか?"}
]
)
print(f"入力トークン数: {token_count.input_tokens}")
```

## 41.9 次のステップ

Claude API の基本を理解できたら、さらに活用の幅を広げましょう。

- [MCP ガイド](#) (ページ 345) — Model Context Protocol を使って Claude に外部ツールを接続する方法
- [Anthropic 公式 API リファレンス](#) — 全パラメータの詳細仕様
- [Anthropic Cookbook \(GitHub\)](#) — 実践的なレシピとサンプルコード集
- [料金プランの比較](#) (ページ 15) — API と `claude.ai` サブスクリプションの違い



---

# AI・Claude 用語集

「トークン」「コンテキストウィンドウ」「ハルシネーション」……Claude や AI の記事を読んでいると、聞き慣れない言葉が次々と出てきます。

このページは、そんな用語を調べるための辞典です。カテゴリ別に整理しているので、知りたい用語を見つけてください。他の記事から「この用語の意味は用語集を参照」とリンクされている場合も、ここに飛んでくることができます。

**このページの使い方:** 右側の目次から知りたい用語を選ぶか、ブラウザの「ページ内検索」(Windows: Ctrl+F / Mac: Cmd+F) で用語を検索してください。

## 基本用語

### AI (人工知能)

**Artificial Intelligence / 人工知能 / AI (エーアイ)**

**一言で:** 人間のように考えたり判断したりする能力をコンピューターに持たせる技術の総称。

コンピューターに「学習する能力」や「問題を解く能力」を与える研究・技術の分野全体を指します。「AI」という言葉はとても広い意味で使われており、チェスを指すプログラムから、会話をこなすアシスタント、画像を認識するシステムまで、様々な技術が AI に含まれます。

Claude や ChatGPT のような会話 AI は「生成 AI (Generative AI)」と呼ばれる AI の一種で、人間の書いたテキストのような新しいコンテンツを生み出すことが得意です。

**関連用語:** [機械学習](#) / [大規模言語モデル \(LLM\)](#)

### 機械学習

**Machine Learning / ML (エムエル)**

**一言で:** 大量のデータを見せることで、コンピューターが自動的に「パターン」を学んでいく技術。

人間がプログラムにルールを細かく教え込む代わりに、大量のデータをコンピューターに見せて、そこからパターンを自分で学ばせる仕組みです。

たとえば「スパムメール判定」なら、大量のスパムメールと普通のメールを学習させることで、新しいメールがスパムかどうかを自動で判断できるようになります。Claude のような AI も、インターネット上の大量のテキストを「学習」することで、自然な文章を生成できるようになっています。

**関連用語:** [深層学習 \(ディープラーニング\)](#)

### 深層学習 (ディープラーニング)

**Deep Learning / 深層学習**

**一言で:** 機械学習の一手法で、人間の脳の神経回路を模した「ニューラルネットワーク」という多層構造を使う技術。

機械学習の中でも特に強力な手法のひとつです。「層（レイヤー）」を何十〜何百層も重ねた複雑な構造（ニューラルネットワーク）を使い、画像認識・音声認識・自然言語処理など幅広い分野で高い精度を実現しています。

Claude をはじめとする現代の大規模言語モデルは、この深層学習技術をベースに構築されています。

**関連用語:** 機械学習 / 大規模言語モデル (LLM)

## 大規模言語モデル (LLM)

**Large Language Model / 大規模言語モデル / LLM (エルエルエム)**

**一言で:** 膨大なテキストデータを学習した、文章の生成・理解を得意とする大規模な AI モデル。

インターネット上の書籍・記事・ウェブページなど、膨大な量のテキストを学習した AI モデルです。「次にどの単語が来るか」を繰り返し予測することで、人間に近い自然な文章を生成できるようになります。

Claude、ChatGPT (OpenAI)、Gemini (Google) は代表的な LLM を活用した AI アシスタントです。「大規模 (Large)」という言葉は、モデルのパラメーター（設定値）の数が数十億〜数千億という規模であることを指しています。

**つまづきポイント:** 「LLM は何でも知っている万能な存在」と思われがちですが、LLM には「学習データに含まれていない最新情報は知らない」「間違った情報を自信満々に答えることがある」という限界があります。詳しくは[ハルシネーション](#)の項目を参照してください。

**関連用語:** トークン / コンテキストウィンドウ / ハルシネーション

## 自然言語処理 (NLP)

**Natural Language Processing / 自然言語処理 / NLP (エヌエルピー)**

**一言で:** コンピューターが人間の言葉（自然言語）を理解・処理・生成するための技術分野。

ここで言う「自然言語」とは、プログラミング言語などの人工的な言語に対して、日本語・英語・中国語などの人間が日常的に使う言葉のことです。コンピューターが人間の言葉を理解し、翻訳したり要約したり、質問に答えたりできるようにする技術の総称を NLP と呼びます。

Claude などの AI アシスタントは、NLP 技術の最先端に立つシステムです。

# Claude 固有の用語

## Claude (クロード)

**Claude**

**一言で:** Anthropic が開発した AI アシスタントの名前。

*Anthropic* が開発・提供する AI アシスタントです。会話・文章作成・分析・コーディングなど幅広いタスクをこなします。Web ブラウザ、デスクトップアプリ、スマートフォンアプリ、API など様々な形で利用できます。

Claude という名前は、claude.ai のほかに Claude Desktop、Claude Code など、Anthropic の AI 製品全体を指すブランド名としても使われます。

**関連用語:** *Anthropic / Haiku / Sonnet / Opus*

## Anthropic (アンソロピック)

### Anthropic

**一言で:** Claude を開発・提供している AI 企業の名前。

2021 年に設立されたアメリカの AI 企業です。「AI の安全な開発」を中心的な使命として掲げており、OpenAI (ChatGPT の開発元) や Google (Gemini の開発元) と並ぶ主要な AI 企業のひとつです。

Anthropic の研究成果である *Constitutional AI* や *RLHF* などは、Claude の安全性・誠実さを支える重要な技術です。

## Artifacts (アーティファクト)

### Artifacts

**一言で:** Claude が作成した文書・コード・HTMLなどを、その場でプレビュー・編集・ダウンロードできる機能。

Claude Web (claude.ai) の機能のひとつです。Claude に「レポートを作って」「表計算の計算式を教える」「簡単なウェブページを作って」などと頼んだとき、その成果物 (アーティファクト) がチャット画面の右側に表示されます。

プレビューを見ながら「もう少し短くして」「見出しを追加して」と修正を依頼できるほか、作成した内容をコピーしたりダウンロードすることもできます。

**類似概念との違い:** 通常のチャット回答はテキストがチャットに表示されるだけですが、Artifacts は成果物を独立したパネルに表示し、インタラクティブに操作できる点が異なります。

## Projects (プロジェクト)

### Projects / プロジェクト

**一言で:** よく使う資料・ルール・背景情報をまとめておき、毎回の会話で自動的に Claude に参照させる機能。

Claude Web と Claude Desktop で使える機能です。たとえば「自社の製品カタログ」「会社のライティングルール」「プロジェクトの仕様書」などのファイルや情報をプロジェクトに登録しておく、そのプロジェクト内で会話するたびに Claude が自動的に参照してくれます。

毎回「この資料を見てください」と貼り付ける手間がなくなり、文脈を共有した状態で継続的に作業できます。

**類似概念との違い:** 通常のチャット会話は会話ごとに記憶がリセットされますが、Projects は複数の会話にまたがって同じ情報 (プロジェクト知識) を参照できます。

## Deep Research (ディープリサーチ)

### Deep Research

**一言で:** Claude がウェブを検索しながら複雑な調査を行い、詳細なレポートを作成する機能。

Claude Web で利用できる機能で、「国内 SaaS 市場のトレンドを調査して」「競合他社の製品を比較してまとめて」のような複雑な調査タスクを Claude に任せられます。Claude が複数のウェブサイトを巡回して情報を集め、最後に引用付きのレポートとして整理してくれます。

通常のチャット回答が数秒で返ってくるのに対し、Deep Research は数分かかることもあります、その分情報の網羅性が高くなります。

## Memory (メモリ)

### Memory

**一言で:** 会話をまたいで Claude が記憶を保持し、ユーザーの好みや情報を次の会話でも覚えている機能。

通常、Claude の会話は終了するたびに記憶がリセットされます（「あなたは誰ですか？」と毎回聞かれる状態）。Memory 機能をオンにすると、Claude は重要な情報（名前、職業、好みの回答スタイルなど）を記憶し、次の会話でもその情報を活用できるようになります。

Claude Web と Claude Desktop の Settings から有効化・確認・編集できます。

**注意:** Memory に保存される内容は設定画面で確認・削除できます。プライバシーが気になる場合は、Settings > Memory で確認してください。

## Cowork (コワーク)

### Cowork

**一言で:** Claude がバックグラウンドで長時間の作業を続けてくれる機能。

Claude Desktop で利用できる機能です。「この 50 ページの PDF を要約して」「この 100 件のデータを分析して」のような時間のかかる作業を Claude に指示して、自分は別の仕事に取りかかることができます。Claude がバックグラウンドで作業を進め、完了すると通知が来ます。

通常のチャットが「その場で返答をもらう」対話形式なのに対し、Cowork は「作業を依頼して待つ」非同期形式です。

## Claude Code

### Claude Code

**一言で:** ターミナル（コマンドライン）や開発環境で動く、エンジニア向けの AI コーディングアシスタント。

ソフトウェア開発者がコードの作成・編集・デバッグ・リファクタリングを自然言語の指示で行えるツールです。ターミナル上で `claude` コマンドを使うか、VS Code などの IDE（統合開発環境）の拡張機能として利用します。

単にコードを書くだけでなく、ファイルの読み書き、Git 操作、テストの実行など、開発作業全体を Claude が自律的に進めてくれます。

**類似概念との違い:** Claude Web・Desktop・モバイルは一般ユーザー向けの汎用 AI アシスタントですが、Claude Code はエンジニアの開発作業に特化したツールです。プログラミングの知識がない方には Claude Web/Desktop をおすすめします。

## MCP (エムシーピー)

### Model Context Protocol / モデルコンテキストプロトコル / MCP

**一言で:** Claude と外部サービスやツールを接続するための共通の「規格（プロトコル）」。

Claude を Google Drive、Slack、GitHub などの外部サービスと連携させるための標準的な接続仕様です。MCP に対応したサービスや拡張機能（MCP サーバー）を Claude Desktop に追加することで、Claude が外部のデータやツールを直接利用できるようになります。



たとえば「Google Drive の MCP サーバー」を追加すると、「マイドライブにある先月の売上データを読んで分析して」というような指示が可能になります。

**具体例:** USB ポートのように、MCP は「Claude と外部サービスをつなぐ共通の差し込み口」だと考えるとイメージしやすいです。

## モデル関連

### Haiku (ハイク)

#### Claude Haiku / Haiku

**一言で:** Claude モデルファミリーの中で最も軽量・高速なモデル。

「俳句 (Haiku)」から名付けられたモデルで、3つのモデルシリーズ (Haiku / Sonnet / Opus) の中で最もシンプルで高速に動作します。シンプルな質問への回答、大量のテキストの素早い処理、コスト重視の用途に向いています。

一般ユーザーが Claude Web で利用する機会は少ないですが、開発者が API を使って大量リクエストを処理する場合などに選ばれます。

**比較:** Haiku < *Sonnet* < *Opus* の順に性能が高くなり、同時にコストも高くなります。

### Sonnet (ソネット)

#### Claude Sonnet / Sonnet

**一言で:** 性能とコストのバランスが最も取れた、Claude の標準的なモデル。

「ソネット (14 行詩)」から名付けられたモデルで、日常的なビジネスタスクから高度な分析・コーディングまで幅広く対応します。Claude Web では多くの場合 Sonnet がデフォルトで選ばれており、ほとんどのユーザーにとって最も使う機会が多いモデルです。

2025 年時点の最新版は **claude-sonnet-4.5** などのバージョン番号付きで提供されています。

### Opus (オーパス)

#### Claude Opus / Opus

**一言で:** Claude モデルファミリーの中で最も高性能な、複雑なタスク向けのモデル。

「オーパス (作品・傑作)」から名付けられた最上位モデルです。複雑な推論、高度な分析、長時間の集中が必要なタスクで特に力を発揮します。一方で処理速度は Sonnet より遅く、API 利用コストも高くなります。

Claude Web では Pro 以上のプランで選択できます。日常的なタスクには Sonnet で十分なことが多く、Opus は「Sonnet では物足りない」と感じた高度なタスクで使うのがおすすめです。

### トークン

#### Token / トークン

**一言で:** AI が文章を処理する際の最小単位。「文字」よりも少し大きな単位。

AI は文章を「文字」ではなく「トークン」という単位で処理します。英語では 1 単語がおおむね 1 ~ 2 トークン、日本語では 1 文字が 1 ~ 2 トークン程度に相当します。

**なぜ重要か:** Claude の利用料金や処理できる文章の長さ（**コンテキストウィンドウ**）は、トークン数で計算されます。たとえば「最大 200,000 トークン」のコンテキストウィンドウなら、日本語で約 10～15 万文字、文庫本 2～3 冊程度の文章を一度に処理できます。

**つまづきポイント:** 日本語は英語よりもトークン数が多くなりやすいため、同じ文章量でも日本語の方が処理コストが高くなります。

## コンテキストウィンドウ

### Context Window / コンテキストウィンドウ

**一言で:** AI が「今の会話」の中で一度に記憶・参照できる情報量の上限。

会話の中で Claude が参照できる「記憶の広さ」です。**トークン**数で表され、たとえば「200,000 トークンのコンテキストウィンドウ」なら、文庫本 2～3 冊分のテキストを一度に処理できます。

**具体例:** 長い PDF ファイルを Claude に読ませて質問するとき、ファイルの内容がコンテキストウィンドウに収まれば Claude は全体を参照して回答できます。ただし上限を超えた部分は参照できなくなります。

**通常の会話との違い:** 会話が長くなるほどコンテキストウィンドウの残量が減っていきます。「最近の回答が的外れになってきた」と感じたら、新しい会話を始めるのがおすすめです。

## Temperature（温度）

### Temperature / 温度 / テンペラチャー

**一言で:** AI の回答のランダム性（創造性）を調整するパラメーター。数値が高いほど多様な回答が出やすくなる。

0～1（または 0～2）の数値で設定します。

- **低い値（例: 0.1～0.3）:** 毎回ほぼ同じ確定的な回答。ファクトチェックや翻訳など、正確さが求められる用途向け
- **高い値（例: 0.7～1.0）:** 回答にバリエーションが出る。アイデア出しやクリエイティブライティングなど、多様な発想が欲しいときに向く

Claude Web では通常ユーザーが直接設定することはありませんが、API を使う開発者はリクエストごとに調整できます。

**料理に例えると:** 「塩加減を調節するつまみ」のようなもの。正確さを求めるなら薄味（低い値）、アイデアを広げたいなら濃い味（高い値）にイメージするとわかりやすいです。

## プロンプト関連

### プロンプト

#### Prompt / プロンプト

**一言で:** AI に対して行う指示・質問・依頼のこと。

Claude に送るメッセージ全体のことをプロンプトと呼びます。「このメールを丁寧な敬語に直して」「来週の会議の議題を 5 つ考えて」のような文章がプロンプトです。

**良いプロンプトのポイント:**

- 何をしてほしいか（タスク）を明確に書く

- 具体的な条件や制約を伝える（「300 文字以内で」「箇条書きで」など）
- 背景情報や目的を伝える（「〇〇向けのプレゼンに使いたい」など）

**関連用語:** システムプロンプト / *Chain of Thought* / *Few-shot*

## システムプロンプト

### System Prompt / システムプロンプト

**一言で:** Claude の振る舞いや役割を設定する「背景指示」で、会話が始まる前にあらかじめ与えられる。

通常のユーザーのメッセージ（ユーザープロンプト）とは別に、Claude の「キャラクター」や「ルール」を設定する特別な指示です。たとえば「あなたはカスタマーサポートの担当者として応答してください。回答は常に丁寧な敬語で、200 文字以内にまとめてください」のような設定をシステムプロンプトに書くことができます。

一般的に Claude Web のチャットでは意識しませんが、企業が Claude を自社サービスに組み込む際や、Claude Desktop の Projects 機能でよく使われます。API を利用する開発者が詳細にカスタマイズするための重要な機能です。

## Chain of Thought（思考の連鎖）

### Chain of Thought / 思考の連鎖 / CoT（シーオーティー）

**一言で:** AI が最終的な答えを出す前に、思考の過程を段階的に示しながら問題を解く手法。

複雑な問題を解くとき、「まず〇〇を考えて、次に××を検討して……」と思考の過程を示しながら答えを出すアプローチです。プロンプトに「ステップバイステップで考えてください」と加えるだけで、Claude が思考過程を示しながら回答してくれます。

数学の文章問題、複雑な意思決定、論理的な推論が必要な場面で特に効果的です。

**具体例:** 「来月の予算配分を考えて」より「来月の予算配分を、収入と支出の現状確認 → 優先事項の整理 → 配分案の提示 の順で考えてください」と書く方が、Claude は丁寧に考えて答えてくれます。

## Few-shot（フューショット）

### Few-shot / フューショット

**一言で:** Claude に「こんな感じで答えてほしい」という例を 2〜3 個見せてから本題を依頼するテクニック。

プロンプトの中に「こういう入力にはこう答える」という例をいくつか示すことで、Claude に期待する回答の形式やスタイルをイメージさせる手法です。「Few（少ない）」の名の通り、数例の例示（例: 2〜5 個）で Claude の回答スタイルを誘導できます。

**例:**

以下のルールで商品名から一言キャッチコピーを作ってください。

例) スマートウォッチ → 「時間を、もっとスマートに」

例) 電動自転車 → 「坂道も、らくらくスイスイ」

では、この商品のキャッチコピーを作ってください:

ノイズキャンセリングイヤホン

## ハルシネーション

### Hallucination / ハルシネーション (幻覚)

**一言で:** AI が事実でない情報を、もっともらしく自信満々に答えてしまう現象。

大規模言語モデルが持つ代表的な欠点のひとつです。Claude が「存在しない論文を引用する」「企業の正しくない設立年度を答える」「実際にはない機能があると説明する」といった形で現れます。回答の文体が流暢で確信を持った書き方をしているため、一見すると正しそうに見えてしまうのが厄介なところ です。

#### 対策:

- 重要な情報 (数字・固有名詞・引用文など) は必ず別の情報源でファクトチェックする
- 「〇〇は本当に正しいですか? 確信はありますか?」と問い直すと、Claude が訂正してくれることもある
- 「情報源を示してください」と追加で依頼する

**つまづきポイント:** Claude を「全知全能の百科事典」として扱うのは危険です。「補助ツール」として使い、重要な判断には必ず人間が確認を入れましょう。

## API 関連

### API (エーピーアイ)

#### Application Programming Interface / API / エーピーアイ

**一言で:** ソフトウェア同士がデータや機能をやり取りするための「接続口」の仕様。

「API」は Application Programming Interface の略で、異なるソフトウェアがお互いに通信・連携するための規約 (ルール) です。

**わかりやすい例:** レストランで「メニューを見て注文をする」行為に例えられます。お客さん (アプリケーション) がウェイター (API) に注文 (リクエスト) を渡すと、ウェイターがキッチン (Claude) に伝えて、料理 (回答) をお客さんに届けてくれます。

Claude の場合、開発者が「Anthropic の Claude API」を通じて自分のアプリやサービスに Claude の機能を組み込むことができます。

**一般ユーザーとの関係:** Claude Web や Claude Desktop のような製品を使うだけなら API を直接操作する必要はありません。API は主に開発者が使うものです。

**関連用語:** [エンドポイント](#) / [API キー](#)

### エンドポイント

#### Endpoint / エンドポイント

**一言で:** API へアクセスするための特定の URL (住所)。

API には複数の機能があり、それぞれに対応した URL (エンドポイント) があります。たとえば Claude API には「メッセージを送信する」エンドポイント (<https://api.anthropic.com/v1/messages>) があります。開発者はこの URL に対して「質問」などのデータを送り、Claude からの回答を受け取ります。

## API キー

### API Key / API キー

**一言で:** API を利用する際の「認証パスワード」。誰がどれだけ API を使ったかを識別するための文字列。

Anthropic の API を使うには、Anthropic のコンソール画面で発行した「API キー」と呼ばれる文字列（例: `sk-ant-api03-...`）が必要です。リクエストのたびにこのキーを添付することで、Anthropic が「誰がどれだけ使ったか」を把握し、利用料金を計算します。

**セキュリティ上の注意:** API キーは「パスワード」と同様に扱ってください。GitHub などのソースコードに直接書いてしまうと、不正利用のリスクがあります。環境変数や秘密管理ツールを使って安全に管理することが重要です。

## レート制限

### Rate Limit / レート制限

**一言で:** 一定時間内に送れるリクエストの上限。過剰な利用を防ぐための仕組み。

API を使いすぎてサーバーに負荷がかかりすぎないように、「1 分間に送れるリクエストは最大〇回まで」「1 日に使えるトークンは最大〇〇万トークンまで」という上限が設けられています。上限を超えるとエラーが返り、しばらく待つ必要があります。

Claude Web のユーザーが日常的に感じる「メッセージ上限」も、広い意味でのレート制限のひとつです。

## ストリーミング

### Streaming / ストリーミング

**一言で:** Claude が文章を生成しながら、リアルタイムで少しずつ画面に表示する方式。

通常の API では Claude が全文を生成し終えてから一気に回答が届きますが、ストリーミング方式では生成しながらリアルタイムで表示します。Claude Web でチャットをすると、回答が少しずつ流れるように表示されますよね。これがストリーミングです。

ユーザーは全文が完成するまで待たなくて良いため、体感上の応答速度が速くなります。

## Function Calling (ツール使用)

### Function Calling / Tool Use / ファンクションコーリング / ツール使用

**一言で:** Claude が外部の関数やツールを呼び出して、リアルタイムの情報を取得したり実際の操作を行う機能。

Claude は基本的に学習済みの知識から回答しますが、Function Calling を使うと「今日の天気を調べる」「データベースを検索する」「カレンダーに予定を追加する」といった外部の機能を Claude が呼び出せるようになります。

開発者が「こういう関数（ツール）が使えるよ」と Claude に教えておくと、Claude が必要に応じてそのツールを使いながら回答してくれます。MCP (*Model Context Protocol*) も、この仕組みを応用したものです。

## セキュリティ・安全性関連

### Constitutional AI（コンスティテューショナル AI）

#### Constitutional AI / 憲法的 AI / CAI（シーエーアイ）

**一言で:** AI が守るべき「原則（憲法）」をあらかじめ設定し、その原則に従って自分の回答を評価・改善させる Anthropic の AI トレーニング手法。

Anthropic が開発した独自の技術です。通常の AI トレーニングでは人間が膨大な数の回答を評価しますが、Constitutional AI では「人間の尊厳を尊重する」「危険な情報を教えない」などの原則リスト（憲法）を AI に与え、AI 自身が自分の回答をその原則に照らして評価・修正します。

これにより、Claude は有害なコンテンツを避け、誠実で役立つ回答をするよう訓練されています。

### RLHF

#### Reinforcement Learning from Human Feedback / 人間のフィードバックからの強化学習 / RLHF（アールエルエイチエフ）

**一言で:** 人間が「この回答は良い / 悪い」と評価し、その評価を使って AI をより良くしていく学習方法。

AI が生成した複数の回答に対して人間が優劣をつけ、「良い回答」をより多く生成するよう AI を訓練する手法です。ChatGPT や Claude など、現代の主要な AI アシスタントの多くがこの手法でトレーニングされています。

人間の価値観や好みに沿った回答ができるよう AI を調整するために使われます。

### セーフティ

#### Safety / AI Safety / セーフティ / AI 安全性

**一言で:** AI が有害なことをしないよう設計・訓練・監視する取り組みの総称。

AI が「嘘をつかない」「危険な情報を教えない」「差別的な発言をしない」「悪意ある目的に使われないようにする」といった安全性を確保するための研究・技術・ポリシーの総称です。

Anthropic はこの「AI Safety（AI 安全性）」を最重要の使命として掲げており、Claude のすべての設計判断においてセーフティを優先しています。

### アラインメント

#### AI Alignment / アラインメント

**一言で:** AI の目標・行動を人間の価値観や意図と一致（アライン）させようとする研究分野。

AI が「役に立つ・無害・誠実」であるように、人間の意図や価値観に沿って動くよう設計する研究分野です。AI が予期しない行動を取ったり、人間の意図に反した目標を追求したりしないようにすることが中心的な課題です。

*Constitutional AI* や *RLHF* はどちらも、このアラインメントを実現するための具体的な技術手法です。

---

## 次のステップ

用語の意味はわかりましたか？次は実際に Claude を使いながら理解を深めてみましょう。

**Claude を使い始めたい方:**

- [Claude とは？— 全体像をつかもう](#) (ページ 3) — 製品の選び方とはじめ方
- [はじめての Claude Web チャット](#) (ページ 23) — ブラウザで今すぐ試す

**プロンプト（指示の出し方）を学びたい方:**

- [プロンプトの基本原則](#) (ページ 239) — 良い指示の書き方を学ぶ

**Claude の安全な使い方を知りたい方:**

- [安全な使い方・AI の限界](#) (ページ 51) — ハルシネーションなどのリスクと対策

**API や技術的な活用をしたい方:**

- [API 入門 — Claude をアプリに組み込む](#) (ページ 357) — 開発者向けの API 解説
- [MCP \(Model Context Protocol\) ガイド](#) (ページ 345) — 外部サービスとの連携方法