

まず、自分の現在地を測る

この講座は、無料の診断ツール「AIユーザーレベルチェック」の結果を土台にして進みます。動画を見る前に、まず一度受けてください。このページでは、開いてから結果を読むまでを、実際の画面で順に説明します。

AIユーザーレベルチェック

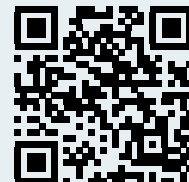
全10問・約2分・登録不要（無料）

画面で見ている人：下のURLをクリック

<https://ai-sozo.com/tools/ai-user-level>

Ctrl（Macは⌘）を押しながらクリックすると、新しいタブで開きます

紙で読んでいる人：右のQRコードをスマホのカメラで読み取る



回答はブラウザの中だけで計算され、サーバーには送られません（診断ページの説明より）。

STEP 1 ページを開いて「はじめる」を押す

上のリンクかQRコードでページを開くと、この画面が出ます。

- 点数で順位をつける診断ではありません。「どこが強く、どこが弱い」の凸凹を見るためのものです。
- 画面下の赤い「はじめる」を押すと、1問目が出ます。

STEP 2 10問に答える

1問ずつ、4つの選択肢から**いちばん当てはまるものを1つ**選びます。選ぶとすぐ次の問題に進みます。

- 右上の「1 / 10」が、いま何問目かです。
- 左上の小さなラベル（例：委任）は、その問題がどの軸を測っているかを表しています。
- 「こうできたらいいな」ではなく、**ふだん実際にやっているやり方**で選んでください。迷ったら、ここ1か月でいちばん多いやり方を選びます。
- 押し間違えたら、画面の下にある「ひとつ前に戻る」で戻れます。

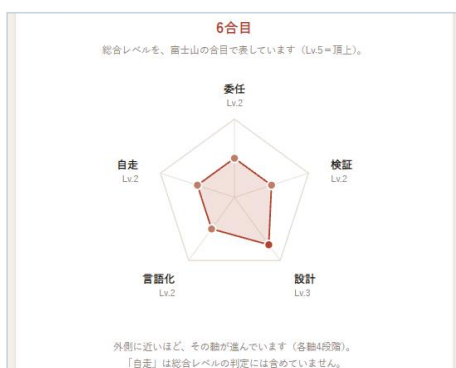
STEP 3 総合レベルを見る



10問目に答えると、結果が出ます。いちばん上が**総合レベル**です。

- 総合レベルはLv.1～Lv.5で、富士山の何合目にいるかの絵でも表されます（Lv.5が頂上）。
- レベルの名前（例：まとめて任せられる段階）と、2行の説明がつきます。
- この画面は回答の一例です。人によってレベルも絵も変わります。

STEP 4 5つの軸の凸凹を見る



五角形のグラフは、軸ごとの進み具合です。**外側に近いほど、その軸が進んでいます**（各軸4段階）。

グラフの下に、同じ内容を表にしたものがあります。いちばん低い軸が、いまの伸びしろです。

軸	何を測っているか	講座で扱う回
委任	どこまでAIに預けているか	第2回 委任を広げる実践
検証	確かめ方が「目」か「仕組み」か	第3回 検証を仕組み化する実践
設計	止めどきと、往復回数の設計	第4回 止めどきを設計する実践
言語化	自分のやり方を説明できるか	第5回 言語化する実践
自走（別枠）	自分が動かなくても進むか	第7回 次の扉：自走

「自走」だけは総合レベルの判定に含まれていません。まだ定番のやり方が固まっていない領域だからです。今すぐ手をつけなくて大丈夫です。

STEP 5

「次の一手」を読む

次の一手は「検証」

確かめ方が、まだ自分の目に頼っている段階のようです。
ここはループの質を決める要であり、いちばん疲れる部分でもあります。

確かめ方を「語が・何を基準に・どうやって」の3点セットで書き出してみてください。
それだけで、指示に移せるようになります。

AIにチェックさせるときは「これで大丈夫？」ではなく「**弱点を3つ挙げて**」。
言い方を変えるだけで、返ってくる厳しさが変わります。

AIも、自分が作ったものには甘くなります。
効くのは別のセッションでチェックさせること。
作った会話から切り離すだけで、お金はかかりません。

表の下の色のついた枠が、**いちばん先に伸ばすとよい軸**と、そのための具体的なアドバイスです。

- この例では「次の一手は『検証』」と出ています。
- 講座は第2回から順に進めるのがおすすめですが、ここに出た軸の回は、特に時間をかけて取り組んでください。
- その下に「自走」についての一言も出ます。

STEP 6

結果を残しておく

自走：入口に立っています
型やテンプレを用意して、動かす手間そのものは減らせている状態です。
この先は「自分が思い出さなくても始まる」段階ですが、まだ定書のやり方が固まっていない領域でもあります。
急ぐ場所ではありません。

最後最後の回答を修正する 最初最初からやり直す

気に入ったら、シェアお願いします

LINE X Facebook URLをコピー

ふければ各SNSもフォローしてください

Xをフォロー

現在地が分かったら、次は手を動かす番です。

関連する記事を読む AI活用セルフ診断もやってみる

結果はどこにも保存されません。あとで見返せるように、残しておきましょう。

- 結果が出ると、回答がページのURLに書き込まれます。「**URLをコピー**」を押してメモしておけば、同じ結果をあとで開けます。
- 最後の1問だけ直したいときは「最後の回答を修正する」、全部やり直すときは「最初からやり直す」を押します。
- 下の記入欄にも書き写しておくのと、講座を終えたあとに比べやすくなります。

記入シート | わたしの現在地

軸の欄には「2 / 4」のように書きます。2回目は、講座を終えたあとにもう一度受けて記入してください。

	1回目（講座を始める前）	2回目（第8回を終えたあと）
受けた日		
総合レベル・何回目		
委任		
検証		
設計		
言語化		
自走		
次の一手		

8回で、ループの外側にいる設計者になる

初級では、自分がループの中の1ステップとして動いていました。この講座では、実行・確認・直し・止めどきを仕組みの側に回し、**自分はその仕組みを設計して、たまに結果を確かめる人になることを目指します。**

回	テーマ	軸	この回で作るもの
1	自己診断	—	診断結果と、いちばん低い軸のメモ
2	委任を広げる実践	委任	どこで確認を挟むかの判断基準
3	検証を仕組み化する実践	検証	判定基準つきの検証スキル
4	止めどきを設計する実践	設計	検証スキルに足す、止めどきの1行
5	言語化する実践	言語化	委任・検証・止めどきを1つにまとめた、人に渡せるドキュメント
6	Claude Codeディレクトリ構造マップ	—	hooksへの配置と、settings.jsonへの登録
7	次の扉：自走	自走	触る場所を分けた、並列運用の試し
8	セキュリティ設定を段階的に固める	—	足した設定と、それが本当に効くかの確認

進め方

- ・第2回から第5回は、上から順に進めてください。言語化（第5回）は、それまでの3回で作ったものがないと書くことがなく、空回りしやすいからです。
- ・各回の書き込みシートは、前の回のシートを使う行があります。全部のシートを1か所にまとめて保管してください。
- ・「外に出る」には2段階あります。第2～5回は、1回の実行の**最中**を手放す段階です（始めるのはまだ自分）。第7回で見る「自走」は、**始めるきっかけ**さえ手放す段階です。

自己診断

「それなりに使えている」の中身は、人によってまるで違います。まず、自分の凸凹を測るところから始めます。

要点

- ・AIユーザーレベルチェックは、順位を出すものではありません。4つの軸（委任・検証・設計・言語化）の凸凹を見て、伸ばす場所を知るためのものです。受け方は、このハンドブックの「はじめに」を見てください。
- ・いちばん低く出た軸が、いまの伸びしろです。
- ・「自走」は総合レベルに含まれません。まだ正攻法が定まっていない領域なので、今すぐ手をつけなくて大丈夫です。
- ・この講座の原則は「自分はループの外側にいる設計者になる」です。各回で、この感覚を思い出してください。

書き込みシート | 第1回 自己診断

いちばん低い軸：

そう思う理由：

最近、実際にそれが起きた作業：

※ 3行目の作業は、第2回でそのまま題材に使います。消さずに保存してください。

委任を広げる実践

委任を広げるとは、全部を投げることではありません。**どこを任せて、どこで確認するかを、先に設計しておくことです。**

要点

- ・任せる粒度は「信頼できるかどうか」ではなく、**間違えたときの損害の大きさ**で決めます。
- ・やり直しが簡単な作業（構成案、言葉での提案など）は、一気に任せてかまいません。
- ・お金や時間がかかり、後戻りしにくい作業（画像・動画の生成、公開、送信など）は、その**直前にだけ確認**を置きます。
- ・正しいチェックポイントの数は決まっていません。同じ「動画づくり」でも、生成コストが高い絵本は場面ごとに、コストが低い教材は最初と最後だけ、というように作業ごとに設計し直します。

書き込みシート | 第2回 委任を広げる実践

選んだ作業：（第1回シートの3行目をそのまま使う）

後戻りしにくい・コストが高いポイント：

そのポイントの手前にだけ確認を置けないか：

※ 第1回のシートと合わせて保管してください。

検証を仕組み化する実践

手順書に「確認すること」と書いてあるだけでは、まだ心がけです。判定基準・自動の発動・直るまでのループがそろって、初めて仕組みになります。

要点

- AIは、自分の出したものを自分だけで見直すのが苦手です。外から「どの基準を満たせていないか」を示すと、大きく改善します（RefineBenchという研究の結果）。
- だから、いちばん大事な仕事は「何をもって合格とするか」を先に言葉にすることです。
- 基準は感想ではなく、確かめられる形で書きます。×「なんとなく読みにくい」→○「一文が60字を超えている」。3～5個から始めます。
- その基準で確認させる指示を、スキルとして保存します。過去の「ダメだった例」で試して、ちゃんと引っかかるかを確かめます。
- この回で作るスキルは「呼べば動く」ところまでです。毎回かならず自動で走らせる方法（フック）は、第6回で扱います。

書き込みシート | 第3回 検証を仕組み化する実践

選んだ作業：

良かった例（3つ）：

ダメだった例（3つ）：

違いを基準にしたもの（3～5個）：

スキル名：

ダメだった例で試した結果： ☐ 引っかった（完成） ☐ 引っかからなかった（基準を書き直す）

※ 第1回・第2回のシートと合わせて保管してください。第4回で、このスキルに止めどきを足します。

止めどきを設計する実践

「直るまでループする」には、いつ止めるかが要ります。止めどきは、直している最中ではなく、始める前に決めます。

要点

- ・止めどきを決めていないと、延々と直し続けるか、結局その場で自分が打ち切るか、のどちらかになります。
- ・止めどきの型は4つです。合格基準／回数上限（「3回まで」など）／頭打ち（前回との差がほとんどなくなったら）／チェックリスト（全項目クリアまで）。
- ・基準がはっきりした作業には合格基準かチェックリスト、感覚が絡む作業には回数上限か頭打ちが向きます。
- ・第3回のスキルの指示文に、1行足します。例：「○回直しても基準を満たさない場合は修正を止め、まだ満たせていない点を報告してください。」
- ・往復が多すぎる場合は、最初に渡す情報（完成イメージ・条件・参考例）を厚くすると、往復そのものが減ります。

書き込みシート | 第4回 止めどきを設計する実践

選んだ止めどきの型（合格基準／回数上限／頭打ち／チェックリスト）：

第3回の検証スキルに足す、止めどきの1行：

試した結果（正しく止まったか）：

※ 第1回～第3回のシートと合わせて保管してください。

言語化する実践

「自分は分かっている」と「人が読んで再現できる」は、まったく別のレベルです。ここまでの設計を、**人に渡せる形**に整えます。

要点

- ・第2～4回で作ったもの（委任の範囲、検証の基準、止めどきの1行）を、1つのドキュメントに集めます。
- ・それぞれの決定に、**なぜそう決めたのか**を1文ずつ添えます。理由がないルールは、読んだ人が自分の状況に合わせて直せません。
- ・書けたら、人か、何も知らない別セッションのAIに渡して、同じ判断ができるか試します。
- ・迷ったときの基準：「このドキュメントだけを渡された人は、自分に一切質問せずに同じ判断ができるか？」。質問が浮かんだら、その答えを書き足します。

書き込みシート | 第5回 言語化する実践

3つの設計をまとめたドキュメントの置き場所（ファイル名）：

「なぜ」を1文で書けなかった項目：

渡してみた相手（人／別セッションのAI）と、出た質問：

※ 第1回～第4回のシートと合わせて保管してください。このドキュメントが第6回の土台になります。

Claude Codeディレクトリ構造マップ

言葉にした設計を、実際の環境に置く回です。いちばん詰まりやすいのは、**hooks/にスクリプトを置くだけでは動かないところ**です。

要点

- CLAUDE.mdは、自然な言葉で書く「人向けの約束事」です。settings.jsonは、JSONで書く「システム向けの設定」です。
- フックは、settings.jsonのhooksキーに「いつ・どのスクリプトを走らせるか」を登録して、初めて動きます。**.claude/hooks/は置き場所にすぎません。**
- CLAUDE.mdに書いたことはお願いなので、AIが守らないこともあります。**必ず守らせたいことは、フックで仕組みにします。**
- permissionsは、allow（許可）・ask（毎回確認）・deny（禁止）の3種類です。
- 設定は、組織の管理設定 → 起動時の指定 → settings.local.json → settings.json → ~/.claude/settings.json の順に強く効きます。効かないときは、この順番のどこかで上書きされていないか疑います。

書き込みシート | 第6回 Claude Codeディレクトリ構造マップ

自分の.claudeフォルダで、この地図のどこに当たるか分からなかったもの：

第5回のドキュメントの中で、hooksに置けそうなもの1つ：

そのhookをsettings.jsonに登録した行（登録した／まだ）：

※ 第1回～第5回のシートと合わせて保管してください。

次の扉：自走

ここから先は、まだ誰にとっても正攻法が定まっていない領域です。焦らず、**入り口だけ**をのぞきます。

要点

- ・一本道のループだけでは、うまくいかないときに「本当に行き止まりか、今の道が悪かっただけか」を確かめられません。
- ・複数の道を同時に試し、いちばん良いものを選ぶ発想が、グラフエンジニアリングの入り口です。
- ・手前の一步は並列運用です。AIの結果を待つあいだに、別のセッションで別の作業を進めます。
- ・**同じファイルを、同時に2つのAIに触らせないこと**。AIは相手が触っていることを知らないので、警告なしに上書きが起きることがあります。触る場所を必ず分け、まずは2つ、多くて3つから始めます。
- ・始めるきっかけ（メールが届いた、時間になった）までAIに任せる状態が「自走」です。今は、そういう方向があると知っておけば十分です。

書き込みシート | 第7回 次の扉：自走

並列で走らせる2つの作業（触る場所が別々であること）：

それぞれが触るファイル（場所）：

並列で試した結果（かかった時間・作業が壊れなかったか）：

※ 第1回～第6回のシートと合わせて保管してください。

セキュリティ設定を段階的に固める

任せる範囲が広がるほど、間違ったときの影響も大きくなります。そして、**書いた設定が本当に効いているかを試すところまでが、設定です。**

要点

- ・固める順番は4段です。土台（サンドボックス）→ 壁（denyルール・機密ファイル・通信先の制限）→ 鍵（確認を全部飛ばすモードを使えなくする設定、フックでの検証）→ 見直し（許可の棚卸し・隔離環境・管理者設定）。
- ・CLAUDE.mdに「このファイルは読まないで」と書くのはお願いします。仕組みで止めるには、settings.jsonのdenyルールに書きます。
- ・ルールがぶつかったときは、deny（禁止）→ ask（確認）→ allow（許可）の順に判断されます。許可を広く出しても、危ない形だけはdenyで止められます。
- ・ドキュメントに書いてあることを、そのまま信じないこと。自分の環境で、実際に動かして確かめます。
- ・全部を一度にやる必要はありません。1つ足したら、効いているかを1つ試す。これを繰り返します。

書き込みシート | 第8回 セキュリティ設定を段階的に固める

自分の環境で、今すぐ足したいdenyルール1つ：

そのルールが本当に効いたか、試した結果（効いた／効かなかった／まだ）：

ドキュメントに書いてあるのに、自分の環境ではまだ確かめていないこと1つ：

※ 第1回～第7回のシートと合わせて保管してください。これで講座は完了です。「はじめに」の記入シートに、2回目の診断結果も書いておきましょう。