Vibe Coder Bootcamp 第1回 — 心構えと、AI の使い方の基本（聞く・できる・渡す）

講師 泉水亮介



1 表紙

- 第 1 回へようこそ。講師の泉水です。今日から、Vibe Coder Bootcamp が始まります。
- 題は、聞く・できる・渡す。AI と作る、最初の一歩。
- 今日持ち帰ってほしい 3 つが、この題に入っています。中身は、あとでちゃんと話します。
- 今日は、オリエンテーションから入ります。まず、みなさんの自己紹介を聞かせてください。僕の自己紹介は、そのあとに。
- それから、この講座がどういう講座か、これから約 1 か月半でどう進んでいくのかを話して、持ち帰ってほしい 3 つに入ります。



2 持ち帰る 3 つ

- ここからが、この講座の芯
- 持ち帰ってほしいのは、3 つの構え。構えは、AI に向き合うときの基本の姿勢のこと
- 聞く、できる、渡す。1 つずつ見ていく
- 1 つ目は「聞く」。分からないことは、分からないまま置かずに、AI に聞く
- 出番は、知らない言葉が出たとき、エラーが出たとき、次に何をすればいいのか分からなくなったとき
- 考え込む前に聞く。この講座で身につけるのは、この聞き方
- 2 つ目は「できる」。プログラミングができない状態から、AI と一緒ならできる
- この講座の目的は、できなかったことが、AI のおかげでできた、という経験を持ち帰ってもらうこと
- たとえば、ログインの仕組みや、データの保存。名前を聞くだけで難しそうなものを、AI と一緒に作る
- 僕自身が、コードを 1 行も書けないまま、最初の AI ボットを作れた経験から始まっている
- 3 つ目は「渡す」。やりたいことと前提を、情報として渡して頼む
- AI が期待を外すとき、足りないのは AI の力ではなく、こちらが渡した情報であることが多い、と僕は考えている
- 出番は、AI の答えが的外れだったとき。思っていたものと違うものが出てきたとき
- 渡す情報は、さっきも出てきた、コンテクスト
- たとえば、「新規事業のアイデアを出して」とだけ頼むと、当たり障りのない答えが返ってくる
- 誰に向けたものか、予算や期間はどれくらいか、どんな背景があるかを渡すと、答えが変わる
- 渡すのは、目的、制約、前提、思い、具体的な状況
- 何を渡すかの目安は、「渡す」の章で整理する
- 今日の道のり。「聞く」は、「聞く」の話と、宿題で確かめる
- 「できる」は、「できる」の話と、僕の実演と、宿題。「渡す」は、「渡す」の話と、実演と、AI の仕組みと、宿題
- 3 つを話したあとに、僕が実演する。実演を見たあとで、AI が文章を作る仕組みを見て、なぜ情報を渡すと答えが変わるのかが分かる
- 最後が宿題。自分の手で AI に頼むのは、宿題
- 約 1 か月半で自分の SNS を作り切ることに向かう、最初の作業になる
- 話の途中で「あ、これだ」と思ったら、それが合図
- 今日の話は、どれもこの 3 つのどれかに戻ってくる
- この帯が、これから、左上にときどき出てくる。では、1 つ目の「聞く」から



3 エラーは、次にやることを教えてくれる

- 作業の途中で赤い文字（エラー）が出ることがある。ぎょっとして、頭に入らなくなる人は多い
- でもそれは、エラーそのものが怖いというより、英語に身構えているだけだと考えている
- 困るのは、エラーが出ることではない。エラーの前で固まってしまうこと
- 画面は説明用の例。文字で命令を打ち込むターミナルで、docs フォルダの中のファイルを見ようとして、名前が合っていない。AI に頼んだ作業の途中でも、こういう画面が出ることがある。この回で、自分で命令を打つことはない
- よく読むと、中学生レベルの英語しか書いていない
- No は「ない」、such は「そのような」、file はファイル、directory はフォルダ。つなげると「そのようなファイルやフォルダはありません」。ただそれだけ
- 英語が得意でなくても大丈夫。僕もまったく得意ではない
- Windows では文言が変わるが、読んでみるとやさしい英語なのは同じ
- 「ファイルがありません」と言われたら、ファイルを作るか、置けばいい
- 名前か場所が違っているだけなので、作る・置く・名前を直す、のどれか
- エラーは、次にやることを教えてくれている
- ほかも同じ
- command not found は、その命令（コマンド）は見つかりません。使おうとした道具が、この PC でまだ見つからない、と教えてくれている
- Permission denied は、許可されていません。その操作をする許可が足りない、と教えてくれている
- 全部覚える必要はない。読めなくても構わない
- 読めなくても構わない。立ち止まらずに、そのまま AI に渡す
- エラーの文字をコピーして貼り、「これは何ですか。どうすればいいですか」と聞く
- 文字が選べるときは、画面を撮るより、コピーして貼るほうが確実。選べないときだけ画面を撮って貼る（撮り方は次のシーンで）
- Cursor や Claude Code がまだ動かない段階で詰まったときは、Web のチャット AI に貼って聞けば大丈夫
- さっきの僕のボットも、エラーが出るたびにそのまま ChatGPT に渡し続けて、動くところまで行った
- 貼る前に、パスワードや、サービスにつなぐための鍵が写っていないかだけ見る（詳しくは次のシーンで）
- AI は、まず意味を教えてくれる。そして、次の一歩
- つづりが違うかもしれない、docs の中の名前を確かめて、もう一度開いてください。答えの中に、次にやることが入っている
- 画面の返事は説明用の例。実際の言い方は毎回少し違う

- AI に言われたとおり、つづりを直して、もう一度や
- 赤い文字が出たら、AI に貼って、次の一歩を聞いて
- エラーは、次にやることを教えてくれるお手紙のよ
  で、エラーの前で固まらなくなる



4 渡すもの 5 つ

- 初日の新人が聞き返しそうなことに、先回りして答える。それが、渡すものの目安になる。
- 新規事業を頼むなら、5 つ。5 つは決まった型ではなく、頼む仕事によって変わる。渡すものを見つけるための目安。
- 1 つ目は目的・ゴール。何を実現したいか、なぜそれが要るか。
- 「お金を儲けたい」だけで終わらせない。会社の 5 年、10 年の計画のなかで、この事業に何を担わせたいのか。そこが書かれていなければ、AI は考慮のしようがない。
- 予算・人・期間。組織のルール。
- 使えるお金がいくらで、いつまでにやらなければいけないのか。言わなければ AI は知らない。
- AI が知らない、業界・社内・経緯。
- この会社は何が強いのか、過去に誰が何を試して、どこで詰まったのか。
- ここが、新人が初日にいちばん知りたいこと。
- 何を大切にしたいか。AI の知ったことではない話。でも、言えば、AI はそれを踏まえた案を出す。
- 僕の経験では、5 つのうち意外に効くのが、この感情や思い。
- 誰に向けて、いつまでに、どんな形で。アウトプットの形も決める。
- 同じ内容でも、営業向けの資料と、役員向けの説明資料と、部署内の発表資料では、作り方が変わる。人が資料を作るときも、相手に合わせて変えるはず。
- AI にも、誰に向けた、どんな形のものかを伝えないと、どれに合わせればよいか分からない。
- 5 つを入れて頼むと、「新規事業を出して」とだけ頼んだときとの違いは、AI が推測で埋める部分が減ること。
- 返ってきやすいのは、どの会社にも言える一般論ではなく、この会社の強みを生かした、100 万円・3 か月に収まる案。



5 ① トークン化

- 文章の AI が文章を返すまでを、4つのステップにまとめる。渡した文章を切れ端に分け（①）、数字の並びにし（②）、どの言葉が大事かを見て（③）、次に来る言葉を1つ選ぶ（④）
- 覚えなくていい。あとで「あれか」とつながれば十分
- あなたが渡す文章が、この4ステップの入口に入っていく。「渡す」が効く理由は、ここにある
- 出てくるのは文章全体ではなく、たった1語。正確には、次に説明する「トークン」（文章を分けた切れ端）1つ分
- その1語を入力の終わりに足して、もう一度①から回す。長い答えは、この一周を何度も繰り返してできている
- 「今日の天気は」なら、晴れ、です、。と1語ずつ足されていく
- AI は、渡された文章をそのままは扱えない。コンピューターは最後は数字にしなければならないので、まず、トークンと呼ばれる切れ端に分ける
- 英語なら単語かそれより小さい断片、日本語なら1文字前後が目安
- 「今日は晴れです」なら「今日」「は」「晴れ」「です」のように
- 切れ方は、意味の区切りではない。よく出てくる文字の並びを、機械的にひとまとめにしたもの
- 切れ方はモデル（AI の本体）ごとに違って、実際には「晴れ」が「晴」と「れ」に割れるモデルもある
- 日本語は同じ内容でも英語よりトークンを多く使いがちだが、新しいモデルほどその差は小さい。AI に一度に渡せる量の上限も、文字数ではなくトークンの数で決まる（あとの「一度に見られる量の上限」で）
- 受講者に文を1つ口で言ってもらって、講師がその場で打つ（受講者は打たない）。切れ端が組み変わるのを見てもらう
- ※ これは AI 本物の区切り方ではなく、ブラウザの単語分割。区切りは本物とは違う（画面にも書いてある）



6 ④ 次の 1 語

- スマホで「いつも」と打つと「お世話になっております」が候補に出る。あの予測変換を、とても大きく、賢くしたものが LLM。僕は「予測変換の化け物」と呼んでいる
- 予測変換との違いは、文章全体や会話全体を読んでから予測すること
- 出すのは 1 語（正確にはトークン 1 つ分）。「今日の天気は」の続きの 1 語を考えている
- 次に来る言葉の候補は、AI が持っている切れ端のすべて。AI は、その 1 つひとつに確率をつける
- 「今日の天気は」の次なら、説明用の例では、晴れ 35%・曇り 25%・雨 20%・雪 10%・そのほか 10%。画面には上位の候補だけ並べて、残りは「そのほか」にまとめてある。実際のモデルの値ではない
- 確率に沿って、1 つ選ぶ。今回は「晴れ」
- もう一度、同じ質問。今度は「曇り」
- いちばん確率が高い「晴れ」が、必ず選ばれるわけではない。晴れは 35% なので、残りの 65% では、ほかの言葉が選ばれる。確率が高い言葉ほど選ばれやすい、という選び方
- 100 回引くと、だいたい確率どおりに散らばる。雪に 10% ついていれば、真夏でも「雪」が選ばれることがありえる
- 確率で選ぶ性質は、使い方によって味方にも注意点にもなる。同じ頼みごとからいくつも案を出させて見比べるなら、毎回違う案が出るのは味方（アイデア出し）
- 数字や事実を正確に答えさせたいときは、必要な情報を渡して、確かめる
- 選ばれた 1 語は、入力の終わりに足される。さっきの輪のとおり、また①から回して、次の 1 語を選ぶ
- 「今日の天気は晴れ」の次は「です」、その次は「。」。長い答えは、この一周を何度も繰り返してできている
- 答えが流れるように出てくるのは、1 語ずつ作っている姿。出力は 1 語ずつ
- ただし内部では、先を見越した準備をしていることもある、と Anthropic の研究（2025 年）で観察されている



7 変えられるのは渡す文章

- AI が「今日の天気は、晴れです」と答えた。でも、窓の外は雨。AI は、今日の天気を知っているわけではない
- 中にあるのは、学習した大量の文章のパターン。その日の実際の天気は、そこには入っていない。「今日の天気は」のあとに続きやすい言葉に、確率をつけているだけ
- 嘘をつこうとしたのではなく、続きやすい言葉を選んだら外れた
- 渡す文章を変えると、確率が動く。「今日の天気は」だけなら、晴れ・曇り・雨・雪…に確率が広がる
- 「東京の 8 月です。今日の天気は」と足すと、雪の確率が下がって、晴れや曇りの確率が上がる（説明用）
- 「天気予報の天気図を添えて。今日の天気は」と渡すと、雨の確率が跳ね上がる
- 実際の天気の情報を渡せば、AI はそれを材料にして、続きを選べる。検索の道具を持つ AI なら、自分で調べにいって、その結果を材料にすることもできる
- ここまでで、次の 1 語が、渡す文章しだいで動くことが分かった。次の 1 語を左右する材料は、大きく 2 つ
- 1 つ目は、学習で決まった部分。AI があらかじめ、大量の文章で「次の言葉を当てる」練習を重ねて身につけた、内部の数字の値。そのあと人が見本や好みを教えて、指示に従う形に仕上げている
- 使っている最中は変わらない。この会話で AI が賢くなるわけではない（会話が今後のモデルの学習に使われるかどうかは別の話で、サービスの設定で決まる）
- 学習に使った文章には締め切りがあって、それより新しいことは、ここには入っていない（あとの「上限と締め切り」で）
- 2 つ目は、あなたが渡した文章。コンテクストとも呼ぶ
- あなたが入力した文章で、同じ会話の中では、ここまでの会話も毎回入力に入る。だから前の話が効く。新しい会話は白紙から始まる。こちらは、毎回、変えられる
- さっき「渡す」の章で見た「平均的で無難な答え」も、この仕組みで説明がつく、と僕は考えている
- 「新規事業を出して」とだけ渡すと、次の 1 語の候補は、どんな新規事業の話にも続けられる言葉に広がる。選ばれる言葉も、誰にでも当てはまる無難なものになりやすい
- 目的・制約・前提・思い・状況を渡すと、候補の確率が、あなたの話のほうへ動く。「渡す」の章で言った「確率をねじ曲げる」は、この動きのこと
- 毎日の会話で、自分の手で変えられるのは、渡す文章。渡した文章は、4 つのステップの入口に入って、それぞれに効く。僕は、こう整理している
- 長いほど切れ端が増える（①）。どんな言葉で書くかで、数字の並びが変わる（②）。大事なことが関係のない話に埋もれると拾いにくい（③）。目的・

制約・前提が書いてあるほど、確率が求める答えの
- 何を渡し、何を渡さないかを選ぶことが、そのまま
- 実際には、サービスが裏で足している指示や、AI が
文章。「渡す」を当たり前にする理由は、この仕組み
- 実演で答えが変わったのも、同じ理由。同じ AI が、
違い、その先の文章も違った
- ここまでを、もう 1 問、二択で確かめる。そのあと



8 今日の 3 つ

- 最後に、今日の 3 つ。最初に見た、聞く、できる、渡す
- この講義の間、ときどき左上にいた、小さな 3 つの札です
- 小さな札が広がって、最初に見た、大きな 3 枚の札に戻りました
- 章の最初の問い、今日いちばん印象に残った場面は、どれでしたか。この 3 枚で振り返ります
- 今日、出てきた場面を、1 つずつ灯していきます
- 聞く。分からないことは、分からないまま置かずに、AI に聞く
- 思い出してほしいのは、エラーの場面。赤い文字が出て、固まりそうになる。でも、そのまま AI に貼ると、意味と、次の一歩が返ってきました
- エラーは、次にやることを教えてくれている。宿題でも同じで、詰まったら、その画面をそのまま AI に見せて、聞きます
- できる。プログラミングができない状態から、AI と一緒ならできる
- 実演を思い出してください。お題の一言が、企画書のファイルになって、そこから、ページやスライドやアプリへ広がりました。その間、僕はコードを 1 文字も書いていません
- 実演で見たことを、宿題で、自分の手で。8 つのファイルが、自分のフォルダにできます
- 渡す。やりたいことと前提を、情報として渡して頼む。渡した分だけ、答えが自分のほうへ寄る
- 仕組みの章で見ました。ひとこと足すだけで、次の 1 語の確率が動く。天気図を渡すと、雨の確率が跳ね上がる
- 渡す章で言った、資料を渡されない新人と同じで、渡した情報で、答えは変わります
- この 3 つを持ち帰って、宿題で使います
- 詰まったら、画面を AI に見せる。8 つのファイルが、自分のフォルダにできる。[仮] と付いた所は、まだ渡していない情報なので、足して頼み直す
- 次の回は、Web アプリ（ブラウザで開いて使うアプリ）が、どんな仕組みで動いているかを見ます