

AI ディベート練習アプリ

設置マニュアル

AI を相手に、質問・反論・Summary & Refute・ジャッジ練習ができるアプリです。生徒は英語で話し、AI から応答と講評を受け取ります。

プログラミング未経験の先生でも設置できるよう、AI に手伝ってもらう方法もあわせて説明します。

VERSION	ISSUED	AUTHOR
v0.8	2026 年 8 月	小林 良裕 / Debate-Class.com

SECTION 00

練習アプリの概要

このアプリは、GitHub に置いたコードを Render で動かし、AI 用の鍵（API キー）を Render だけに保管して使います。鍵をコードに書かないことが、安全に運用するうえで最も大切な点です。

このアプリでできること

生徒が AI を相手に、次の 4 つのモードを練習できます。

- Attack — AI の立論を聞き、質問して反論する
- Defend — AI の質問に答え、自分の主張を守る
- Judge — 肯定側と否定側を比べて判定する
- Summary & Refute — 相手を要約し、反論し、自分の議論を出す

質問は音声で読み上げられ、答えはキーボード入力でもマイク録音でも入力できます。最後に講評が表示されます。

配布しているコードについて

このマニュアルとあわせて配布しているコード一式は、自分が稼働させている練習アプリから、生徒ごとの個別 ID 発行のしくみと、練習記録を保存するしくみを取り除いたものです。

取り除いた理由は、設置を簡単にし、個人情報を扱わずに試していただけるようにするためです。データベースの用意も、個人情報の管理も必要ありません。

この配布版はゲスト専用です。パスワードを知っている人だけが入室でき、誰が何を書いたかは記録されません。

もうひとつ、稼働させているアプリには「Match（AI と 1 対 1 で対戦する）」という練習を後から足してあります。こちらも配布版には含めていません。作り方は第 12 章でご説明します。

記録や個別 ID が必要になったら

取り除いた機能は、AI に依頼して追加することもできます。第 12 章の作り方と、第 1 章のプロンプトをお使いください。

ただし、生徒の氏名や練習内容を保存する場合は、保管場所と保存期間について、学校の個人情報の取り扱い方針を必ずご確認ください。

設置の全体像

章	内容	目安
01	AI に設置を手伝ってもらう	—
02	アカウントを用意する	20 分
03	API キーを取得する	15 分
04	コードを GitHub へ置く	15 分
05	Render で Web Service を作る	20 分
06	Environment Variables を設定する	10 分
07	同時に何人まで使えるか	—
08	公開後の動作確認	10 分
09	授業での運用と音声確認	—
10	トラブル対処	—
11	費用・安全・プライバシー	—
12	アプリを広げる	—
13	最終チェックリスト	—

初めての方へ

はじめから順に読む必要はありません。まず第 1 章を読み、AI を相談相手として用意してから、第 2 章以降を進めるのが近道です。

分からない画面が出たら、その都度 AI に聞けば止まらずに進められます。

SECTION 01

AI に設置を手伝ってもらおう

この設置作業は、Claude や ChatGPT に相談しながら進めると、はるかに楽になります。分からない画面が出たとき、その場で質問できるからです。

この章では、AI への頼み方と、そのまま貼り付けて使える文章を用意しました。

準備：AI に資料を渡す

まず、AI に「何を設置しようとしているか」を分かってもらいます。次の 2 つを渡してください。

1. このマニュアル（PDF）を、そのままアップロードする
2. 配布コードの ZIP を展開し、中のファイルをアップロードする

ファイルの渡し方

Claude も ChatGPT も、画面のクリップのようなボタンからファイルを添付できます。

ZIP のまま渡せない場合は、展開して `server.js`、`index.html`、`package.json`、`README.md` を個別に添付してください。

本物の API キーが入ったファイル（`.env`）は渡さないでください。配布物の `.env.example` には本物の鍵は入っていません。

プロンプト 1：全体像をつかむ

設置を始める前に、これから何をするのかを説明してもらいます。

（以下をそのままコピーして貼り付けてください）

添付したのは、英語ディベート練習アプリの設置マニュアルとコード一式です。

私はプログラミングの経験がない教員です。

GitHub と **Render** を使って、このアプリを自分の環境に設置したいと思っています。

これから何をするのか、全体の流れを 5 つくらいの段階に分けて、専門用語をできるだけ使わずに説明してください。

各段階で、私が実際に何をクリックするのも教えてください。

プロンプト 2：詰まったときに助けてもらう

操作の途中で分からなくなったら、いま見えている画面を伝えて聞きます。

（そのままコピーして貼り付け、[] の中を書き換えてください）

添付したマニュアルの手順で作業しています。

いま [GitHub / Render] の画面にいます。

画面には [見えているボタンや文字をそのまま書く] と表示されています。

マニュアルの [第o章] まで進みました。

次に何をすればよいか、クリックする場所を具体的に教えてください。

英語の表示は、日本語の意味も添えてください。

画面の文字は、そのまま書き写してください

要約せずに、見えているボタン名や見出しをそのまま伝えるほど、AI は正確に答えられます。

スクリーンショットを貼れる場合は、画像を添付するのが最も確実です。

プロンプト 3：エラーの原因を調べてもらう

赤い文字が出た、アプリが開かない、といったときに使います。

（そのままコピーして貼り付け、[] の中を書き換えてください）

添付したマニュアルの手順で設置しましたが、うまく動きません。

状況： [何をしたら、何が起きたか]

表示されたメッセージ：

[エラーの文章をそのまま貼り付ける]

原因として考えられることを、可能性の高い順に挙げてください。

それぞれについて、確認する場所と直し方も教えてください。

わたしはプログラミングの経験がないので、専門用語には説明を添えてください。

AI に聞くときの心得

- 一度に多くを聞かず、いま困っている 1 つに絞ると、答えが正確になります

- AI の回答が実際の画面と合わないときは、「画面にはそのボタンがありません」と伝えると訂正してくれます
- API キーやパスワードそのものを AI に貼り付けしないでください。「キーを設定した」と伝えるだけで十分です
- AI の説明が古い場合があります。実際の画面の表示を優先してください

作業のあとにも使えます

このアプリは、AI に頼んで機能を追加することもできます。

「練習した内容を保存する機能を追加したい」「質問の数を変えたい」といった相談も、同じようにコードを渡して依頼できます。

SECTION 02

アカウントを用意する

4 つのサービスへ登録します。音声を使わない場合だけ、OpenAI は省略できます。

サービス	役割	必要度
GitHub	コードを保管する	必須
Render	アプリを公開する	必須
Anthropic	ディベートの会話・講評	必須
OpenAI	音声認識・AI 音声読み上げ	音声を使う場合

登録時の注意

- 学校の管理者アカウントではなく、ご自身のアカウントで作業されることをおすすめします
- パスワードは使い回さず、可能なら二段階認証を設定してください
- Anthropic と OpenAI は支払い方法の登録が必要です

公式入口

GitHub	https://github.com/
Render	https://dashboard.render.com/

Anthropic <https://console.anthropic.com/>

OpenAI <https://platform.openai.com/>

SECTION 03

API キーを安全に取得する

最初に必ずお読みください

API キーは、銀行の暗証番号と同じ性質のものです。他人に渡ると、あなたの支払いで AI が使われます。

コード、README、ZIP、メール、SNS へ絶対に書かないでください。AI に貼り付けることも避けてください。

取得画面を離れると二度と表示されません。その場で安全な場所に保存してください。

Anthropic（必須）

1. <https://console.anthropic.com/> にログインします
2. API Keys の画面を開きます
3. Create Key を押し、名前を付けます（例：debate-trainer）
4. 表示された文字列をすべてコピーし、安全な場所に保存します

OpenAI（音声を使う場合）

1. <https://platform.openai.com/api-keys> にログインします
2. Create new secret key を押します
3. 表示された文字列をすべてコピーし、安全な場所に保存します

利用上限を先に決める

どちらのサービスも、月ごとの上限額を設定できます。想定外の請求を防ぐため、設置前に必ず設定してください。

SECTION 04

配布コードを GitHub へ置く

GitHub は、コードを保管しておく場所です。ここに置いたコードを、次の章で Render が読み取って動かします。

1. ZIP を展開する

1. 配布 ZIP をパソコンに保存します
2. ファイルを右クリックし、「すべて展開」を選びます
3. 展開先のフォルダを開き、中身を確認します

中には次のファイルが入っています。

ファイル	役割
index.html	画面とブラウザ側の動作
server.js	AI・音声 API への安全な中継
package.json	Node.js の起動設定
README.md	設定・運用の説明
.env.example	環境変数の見本。本物の鍵は書かない
.gitignore	秘密の鍵を GitHub から除外する設定
LICENSE	配布条件 (MIT)

2. GitHub で repository を作る

repository（リポジトリ）は、1つのアプリのためのフォルダのようなものです。

1. <https://github.com/> を開き、ログインします
2. 画面右上の「+」を押し、New repository を選びます
3. Repository name に名前を入れます（英数字とハイフンで。例：my-english-app）
4. Public（誰でも見られる）か Private（自分だけ）を選びます。基本は Private で構いません

5. 下の「Add a README file」などのチェックは、すべて外したままにします
6. 緑色の Create repository を押します

チェックを外す理由

README などを最初に作ると、あとからアップロードするファイルと重なって分かりにくくなります。

空の状態から始めるほうが確実です。

3. ファイルをアップロードする

作成直後の画面に、次のような案内が出ます。

Quick setup – if you’ve done this kind of thing before
...or upload an existing file

1. その中の uploading an existing file というリンクを押します
2. （この画面を閉じてしまった場合は、Add file → Upload files を選びます）
3. 展開したフォルダを開き、中のファイルをすべて選びます
4. 選んだものを、GitHub の画面の点線の枠へドラッグして落とします
5. 画面を下までスクロールします
6. 緑色の Commit changes を押します

ここが最もつまずきやすい場所です

ドラッグしただけでは保存されません。必ず画面の一番下までスクロールして、緑色の Commit changes を押してください。

これを押さずに画面を閉じると、ファイルは1つも保存されていません。

4. 正しく置けたか確かめる

repository のトップ画面に戻ったとき、次のように見えていれば成功です。

- server.js や package.json が、一覧にそのまま並んでいる

フォルダの二重化に注意

repositoryを開いたとき、フォルダを1つ開かないとserver.jsが見えない状態は誤りです。
その場合はフォルダごとアップロードしてしまっています。ファイルを削除し、フォルダの「中身」を選び直してアップロードしてください。

隠しファイルが見えないとき

.env.exampleと.gitignoreは、点で始まるため、Windowsのエクスプローラーでは既定で表示されません。ドラッグしたつもりでも、この2つだけ漏れることがあります。

表示するには、エクスプローラーの「表示」タブで「隠しファイル」にチェックを入れます。

それでも難しい場合は、GitHubの画面で直接作れます。Add file → Create new fileを選び、ファイル名に.gitignoreと入力し、中身を貼り付けてCommit changesを押します。

SECTION 05

Render で Web Service を作る

Renderは、GitHubに置いたコードをインターネット上で動かし、URLを発行してくれるサービスです。

1. Render にログインする

1. <https://dashboard.render.com/>を開きます
2. 初めての場合は、GitHubアカウントでのログインを選ぶと、あとの接続が楽になります

2. Web Service を作る

1. 画面右上のNew +を押します
2. メニューからWeb Serviceを選びます
3. Git Providerの欄でGitHubを選びます

3. GitHub とつなぐ

ここが最もつまずきやすい場所です。RenderにGitHubを見る許可を与える必要があります。

1. Connect GitHub（または Configure account）を押します
2. GitHub の画面に移り、許可を求められます
3. All repositories（すべて）か、Only select repositories（選んだものだけ）を選びます
4. 後者を選んだ場合は、先ほど作った repository を一覧から選びます
5. Install / Save を押すと、Render の画面に戻ります

repository が一覧に出てこないとき

Render にその repository を見る許可が与えられていません。

一覧の近くにある Configure account（または Edit permissions）を押し、GitHub の画面で repository を追加してください。

Private の repository も、許可すれば使えます。

4. 設定を確認する

repository を選ぶと、設定画面が開きます。次を確認してください。

項目	設定	補足
Name	好きな名前	この名前が URL の一部になります
Region	Singapore など	近い場所を選ぶと表示が速くなります
Branch	main	自動で入ります
Root Directory	空欄のまま	コードを直下に置いた場合
Runtime / Language	Node	自動で判定されます
Build Command	npm install	自動で入ります
Start Command	npm start	ここは必ず確認してください
Instance Type	Free から開始可	あとから変更できます

PORT は自分で追加しません

PORT は Render が自動で設定します。次の章の Environment Variables に、自分で追加する必要は

項目	設定	補足
ありません。		

5. 環境変数を入れてから作成する

作成画面の下のほうに Environment Variables の欄があります。ここで先に鍵を入れておくと、あとから設定し直す手間が省けます。

入れる内容は次の章のとおりです。この場で入れない場合は、作成後に左メニューの Environment から追加できます。

1. 画面下の Create Web Service（または Deploy Web Service）を押します

6. Deploy の様子を見る

作成すると、自動的にログの画面に切り替わります。文字が次々に流れますが、読む必要はありません。

表示	意味
Building / In progress	準備中です。数分かかります
Live（緑）	成功。アプリが動いています
Failed（赤）	失敗。ログに原因が出ています
Deploy succeeded	成功したときにログの最後に出ます

Live になったら、画面上部に表示されている URL を押してください。アプリが開きます。

うまくいかないとき

Failed と出る

- ログを上へさかのぼり、赤い文字の行を探します
- その行をコピーし、AI に貼り付けて原因を尋ねてください（プロンプト 3 をお使いください）
- 多いのは、ファイルの置き方の誤り（フォルダの二重化）です

画面は出るが、パスワードが通らない

- Environment に APP_PASSWORD が登録されているか確認します
- 環境変数を変更した後は、Deploy が終わるまで反映されません

しばらく使わないと、最初の表示が遅い

- 無料プランでは、アクセスがないとアプリが休止します
- 授業の数分前に一度開いておくと、生徒がすぐ使えます

更新するとき

GitHub のファイルを新しいものに置き換えると、Render が自動で再 Deploy します。Environment Variables は保たれるため、鍵の再設定は不要です。

自動で始まらない場合は、画面右上の Manual Deploy → Deploy latest commit を押してください。

SECTION 06

Environment Variables を設定する

Render の画面左の Environment を開き、次の値を登録します。Add Environment Variable を押し、Key（名前）と Value（値）を入れます。

必須の 2 つ

名前 (Key)	値 (Value)
ANTHROPIC_API_KEY	Anthropic で取得した本物の API キー
APP_PASSWORD	生徒に配る、長く推測されにくいパスワード

音声を使う場合

名前	既定値	説明
OPENAI_API_KEY	(必須)	OpenAI で取得した本物の API キー

名前	既定値	説明
ANTHROPIC_MODEL	claude-haiku-4-5-20251001	会話・講評に使うモデル
STT_MODEL	gpt-transcribe	録音した英語を文字にするモデル
TTS_MODEL	gpt-4o-mini-tts	英語を読み上げるモデル
TTS_VOICE	marin	読み上げる声（cedar にも変更可）
TTS_INSTRUCTIONS	（既定文あり）	読み方の速さや雰囲気指定する英文
TTS_RESPONSE_FORMAT	mp3	音声の形式

STT_MODEL 以下は設定しなくても、上の値が既定で使われます。

モデル名は将来変わることがあります

AI のモデル名（gpt-transcribe など）は、各社の更新にともなって変わることがあります。

設置した時点では正しくても、年月が経つと使えなくなることがあります。

うまく動かなくなったときは、公式の一覧で最新のモデル名を確認し、Render の環境変数で指定し直してください。コード自体を書き換える必要はありません。

同時に使う人数に関わる設定

名前	既定値	説明
RATE_LIMIT_AUDIO	10	1 分あたりの音声の利用回数
RATE_LIMIT_AI	30	1 分あたりの AI 応答の回数

同じ回線から 5～6 人までなら、変更は不要です。1 クラスで一斉に使う場合は第 7 章をご覧ください。

パスワードの決め方

- 学校名や年度など、推測されやすい語だけにしないでください
- 生徒に口頭で伝えられる程度の長さにします

- 年度や学期ごとに変更することをおすすめします

保存を忘れずに

入力後、Save Changes を押してください。押すと自動的に再 Deploy が始まり、2～3 分で反映されます。

SECTION 07

同時に何人まで使えるか

生徒が一斉に使うと、短い時間にたくさんの通信が起こります。アプリには、使いすぎを防ぐための回数制限を入れてあります。人数によっては、この設定を変える必要があります。

利用人数ごとの設定

使い方	同時に使う人数	設定変更
部活動・課外活動・個人練習	5～6 人まで	不要
少人数の授業・グループ学習	10 人前後	変更をおすすめします
1 クラスで一斉に	20～50 人	必要です

なぜ人数で変わるのか

この制限は、同じインターネット回線からの利用回数を 1 分ごとに数えています。学校のネットワークから使うと、生徒が何人いても、外からは同じ 1 つの住所（IP アドレス）に見えます。

つまり「1 人あたり何回」ではなく、「学校全体で何回」という数え方になります。

生徒 1 人が 1 問答えるあいだに、読み上げで 1 回、録音の文字起こしで 1 回、あわせて音声で 2 回使います。1 問に 2 分かかるとすると、1 人あたり 1 分間に約 1 回です。既定値は 1 分間に 10 回なので、同時に使えるのは 5～6 人が目安になります。

生徒それぞれの携帯回線を使う場合

生徒が学校の Wi-Fi ではなく、各自の携帯回線（4G・5G）で使うときは、IP アドレスが別々になります。

この場合、上の人数の目安はあてはまりません。

上限に達すると、どうなるか

生徒の画面に「短時間の利用回数が上限に達しました。少し待ってからもう一度お試しください。」と表示されます。1分待てば、また使えるようになります。アプリが壊れるわけではありません。

クラスで使うときの設定

Render の Environment を開き、次の2つを追加してください。

名前 (Key)	既定値	1 クラス (50 人) の目安
RATE_LIMIT_AUDIO	10	120
RATE_LIMIT_AI	30	100

数字を大きくすると、その分だけ多くの人と同時に使えます。0を入れると制限しません。

入力したら Save Changes を押してください。再 Deploy が始まり、2〜3分で反映されます。

緩めるときに、必ずあわせてやること

この制限は、使いすぎと、いたずら目的の大量アクセスを防ぐためのものです。緩めるほど、API 利用料が増える可能性が高くなります。

Anthropic と OpenAI の月ごとの上限額を、先に必ず設定してください。

授業が終わったら、既定値に戻しておくで安心です。

AI 各社にも、別の上限があります

アプリ側の設定とは別に、Anthropic と OpenAI にも、1分あたりのリクエスト数の上限があります。

- Anthropic は、Claude Haiku で毎分 1,000 回まで（利用状況によるランクが、Start ティアの場合）。50 人が同時に使っても十分に足ります

- OpenAI の音声機能は、契約の段階によって上限が変わります。50 人規模なら毎分 100～150 回あれば足ります

ご自身の上限は、それぞれのコンソール画面で確認できます。

Anthropic <https://platform.claude.com/settings/limits>

OpenAI <https://platform.openai.com/settings/organization/limits>

初めて大人数で使う前に

AI のサービスは、利用がとつぜん急に増えると、上限に達していなくても一時的に断られることがあります。

授業の数日前に、数人で一度動かしておくとお安全です。

Render の無料プランはアクセスがないと休止するため、授業の 10 分前に一度 URL を開いておいてください。

Render のプランについて

無料プランでも動きますが、1 クラスで一斉に使うと、表示が遅くなることがあります。また、しばらく使わないと休止し、次の人が 50 秒ほど待たされます。

授業で定期的に使うのであれば、有料の Starter プラン（月 7 ドル程度）にすると、休止がなくなり、処理も速くなります。

SECTION 08

公開後の動作確認

確認すること	結果
パスワードを入力して入室できる	
モードを選んで練習を始められる	
AI の返答が表示される	
「▶ Listen」で最後まで聞こえる	
同じ音声をもう一度再生できる	

確認すること	結果
キーボードで答えを送信できる	
マイク録音が文字になる	
最後に講評が表示される	
スマートフォンでも同じ確認ができる	

SECTION 09

授業で使う場合の運用と音声確認

授業前

- 先生ご自身が一度、最初から最後まで通してやってみます
- 学校のネットワークでアプリの URL が開けるか確認します
- マイクの使用許可が求められることを、生徒に伝える準備をします
- イヤホンを使う場合は、事前に準備の指示をしておきます

生徒に伝えること

- パスワードは他の人に教えないこと
- 氏名・住所・連絡先などの個人情報は入力しないこと
- 聞こえる声は AI が作った音声であり、人間の録音ではないこと
- AI の講評には誤りがあること。最終的な判断は先生が行うこと

音声機能の確認

Deploy 後、AI の発言が表示されたら「▶ Listen」を一度押してください。最後まで再生され、同じ音声をもう一度再生できることを確認します。

SECTION 10

トラブル対処

困ったときは、第 1 章のプロンプト 3 をお使いください。エラーの文章を AI に貼り付けると、原因を調べてもらえます。

音声途中で止まる

1. YouTube、動画、オンライン会議など他にネット回線を利用している機能があれば、止めます
2. 同じ Listen をもう一度押します
3. Bluetooth を外し、本体スピーカーか有線イヤホンで試します
4. 別のブラウザ、別の PC、スマートフォンで比較します

同じ回線で動画を止めると改善する場合、インターネット回線だけでなく、PC の処理、ブラウザ、Bluetooth ・音声機器の競合も考えられます。

マイクが反応しない

- ブラウザがマイクの使用許可を求めているか確認します
- URL が https:// で始まっているか確認します。http では使えません
- 端末の設定でブラウザにマイクの許可が与えられているか確認します

「短時間の利用回数が上限に達しました」と出る

同じ回線から、短い時間に多くの人が使ったときに出ます。1 分待てば、また使えるようになります。

- 数人で使っていて出る場合は、通信が重なっただけです。少し待ってください
- クラス全体で使っていて何度も出る場合は、第 7 章の設定を変えてください
- 心当たりがないのに何度も出る場合は、パスワードが外部に漏れていないか確認してください

入室できない

- パスワードの大文字・小文字、全角・半角を確認します
- Render の Environment に APP_PASSWORD が登録されているか確認します

- 環境変数を変更した後は、Deploy が終わるまで反映されません

SECTION 11

費用・安全・プライバシー

費用

- Render は無料プランから始められます
- Anthropic と OpenAI は使った分だけ費用がかかります
- 1 回の練習でかかる費用はごくわずかですが、人数分積み上がります
- 必ず、月ごとの上限額を設定してください（10～20 ドル程度で十分です）

AI への指示文はサーバーの中にあります

このアプリは、AI への指示文（プロンプト）をサーバーの中に固定しています。ブラウザから送るのは「どの練習か」という名前と、論題や利用者の発話などの項目だけです。

- 指示文そのものを、ブラウザから差し替えることはできません
- 用意されていない練習名を指定すると、サーバーが断ります
- 受け取る項目は、種類と文字数を制限しています

なぜこうしているか

パスワードを知っている生徒が、ブラウザの開発者向け画面を使って、このアプリを「何でも答える汎用の AI 窓口」として使ってしまう可能性を防ぐためです。

API キーそのものはブラウザに渡していないため見られませんが、指示文を自由に送れる作りだと、先生の契約で AI を使われてしまいます。

あわせて、AI 各社の管理画面で月ごとの上限額を必ず設定してください。

このサンプルが保存しないもの

前述のとおり、配布版は運用中のアプリから個別 ID 発行と練習記録の保存を取り除いてあります。したがって、次のものはどこにも残りません。

- 生徒ごとの個別 ID

- 練習した本文
- 評価・講評の内容
- 誰が何回練習したかの記録

記録を残したい場合

練習内容を保存する機能は設けていませんが、AI に依頼して追加することもできます。

その際は、保存する内容と保管場所について、学校の個人情報の取り扱い方針を必ずご確認ください。

外部の API へ送るもの

- ディベートの文章 → Anthropic API
- 録音データと、読み上げる文章 → OpenAI API

利用にあたって、各学校のきまりに従ってください

AI の講評や評価には誤りがあります。生徒への最終的な評価は、必ず先生が行ってください。

各サービスの料金・規約・モデルの提供状況は変更されます。授業で使う前に、公式資料と学校のルールを確認してください。

SECTION 12

アプリに追加機能を加える

配布版をそのまま使うだけでなく、練習機能を足したり、別のアプリを作ったりすることもできます。ここでは二つの広げ方をご紹介します。

広げ方その 1：配布版に練習機能を足す

配布版には、Attack・Defend・Judge・Summary & Refute の 4 つの練習が入っています。ここに新しい練習を足せます。

作例：1 対 1 の対戦モード

自分が稼働させているアプリには、配布版に無い「Match」という練習を足してあります。

AI と 1 対 1 で、立論 → 相手からの質疑 → こちらからの質疑 → 反論 → リプライまで行い、最後に勝敗の判定と 6 項目の採点が出ます。

配布版に対して、画面を 1 枚と、AI への指示を 4 つ足しただけで作れました。

足すときに手を入れる 2 か所

このアプリは、AI への指示文をすべて `server.js` の中に置いています（第 11 章）。そのため、新しい練習を足すときは次の 2 か所だけを直します。

1. `server.js` の `DEBATE_TASKS` に、新しい練習を 1 つ足す（AI への指示文と、生徒側の入力 of 組み立て方を書きます）
2. 画面から、その練習を名前と呼ぶ `callClaude({ task:"新しい練習の名前", vars:{ 論題や生徒の発話 } })`

画面側に指示文を書かないでください

指示文を画面（`index.html`）に書くと、第 11 章で説明した「何でも答える汎用の AI 窓口」として使われる状態に戻ってしまいます。

新しい練習を足すときも、指示文は必ず `server.js` の中に置いてください。

AI に頼むときの指示文

Claude や ChatGPT に、配布版のコード一式を添付したうえで、次のように伝えてください。

そのままコピーして貼り付け、[] の中を書き換えてください

添付したのは、AI を相手に英語ディベートを練習するアプリのコードです。
このアプリに、新しい練習をひとつ足したいと思っています。

- ・練習の名前 : [例：ミニ立論]
- ・生徒がすること : [例：30 秒で自分の立場を述べる]
- ・AI がすること : [例：良かった点と直す点を日本語で返す]
- ・画面の流れ : [例：論題を選ぶ → 話す → 講評]

このアプリは、AI への指示文を `server.js` の `DEBATE_TASKS` の中に置く作りになっています。新しい練習も必ず同じ形にしてください。

画面側（`index.html`）には指示文を書かないでください。

次の 2 つを、そのまま貼り付けられる形で教えてください。

1. `server.js` の `DEBATE_TASKS` に足すコード

2. `index.html` から呼び出すコード

この 2 つのコードを追加したファイルもそれぞれください。

足したあとの確認

- 新しい練習が最初から最後まで動くか、先生ご自身で一度通してみます
- もとからある 4 つの練習が、これまでどおり動くかも確かめます
- うまくいったら、GitHub にアップロードして保存しておきます
- 動かなくなったら、GitHub の履歴から前の状態に戻せます

広げ方その 2：ゼロから別のアプリを作る

題材そのものを変えて、別のアプリを作ることもできます。

先に決める 7 項目

1. 誰が使うか（学年・英語力）
2. 何を練習するか（話す題材）
3. AI はどんな役を演じるか
4. 1 回の練習は何往復か
5. どんな講評を返すか
6. 音声を使うか
7. 記録を残すか

AI へ最初に渡す技術条件

配布版のコードを添付し、次のように伝えると、同じ土台で別のアプリを作れます。

そのままコピーして貼り付け、[]の中を書き換えてください

添付したのは、AIを相手に英語を練習するアプリのコードです。
この作りを参考に、別の題材のアプリを作りたいと思っています。

- ・使う生徒：[学年・英語のレベル]
- ・話す題材：[例：修学旅行の説明／部活動の紹介]
- ・AIの役：[例：訪問先で出会った人]
- ・往復数：[例：8問、途中で終了することも可能に]
- ・講評：[例：良かった点と直す点を日本語で]

技術の条件は元のコードと同じにしてください。

Node.js (Express は使わない)、Render で動かす、API キーは環境変数から読む、
AI への指示文は `server.js` の中に置く、記録は保存しない、
パスワードで入室する、という点は変えないでください。

まず、変更が必要なファイルと箇所を教えてください。

小さく作って確かめる

- ・一度にすべてを変えず、1 つずつ試します
- ・動いた状態を GitHub に保存してから、次の変更に進みます
- ・うまくいかなかったら、GitHub の履歴から前の状態に戻せます

SECTION 13

最終チェックリスト

確認すること	結果
GitHub に必要なファイル（基本 7 つです）が置かれている	
本物の API キーが GitHub に書かれておらず、ファイル内にもない	
Render の Deploy が Live になっている	
ANTHROPIC_API_KEY を Render 上で登録した	
APP_PASSWORD を Render 上で登録した	

確認すること	結果
音声を使う場合、OPENAI_API_KEY を Render 上で登録した	
Anthropic と OpenAI に上限額を設定した（目安は月額 10～20 ドル）	
PC で最初から最後まで動いた	
スマートフォンでも動いた	
同時に使う人数を確かめ、必要なら回数の設定を変えた	
生徒に伝える注意事項を決めた	

更新するとき

GitHub のファイルを新しいものに置き換えると、Render が自動で再 Deploy します。Environment Variables は保たれるため、鍵の再設定は不要です。

SECTION 14

参考リンクと改訂情報

公式資料

GitHub <https://github.com/>
 Render <https://dashboard.render.com/>
 Anthropic Console <https://console.anthropic.com/>
 OpenAI API Keys <https://platform.openai.com/api-keys>

配布ページ

AI 練習アプリの作り方 <https://debate-class.com/ai-debate-app-guide/>

改訂履歴

版	時期	内容
v0.8	2026 年 8 月	配布版に練習を足す方法を第 12 章に追加しました（作例：1 対 1 の対戦モード）。
v0.7	2026 年 8 月	AI への指示文をサーバー側に固定し、汎用の AI 窓口として使われな

版	時期	内容
		いようにしました。
v0.6	2026 年 8 月	AI に設置を手伝ってもらう章を新設。GitHub と Render の操作説明を詳しくしました。配布コードが運用版から個別 ID 発行・練習記録保存を取り除いたものである旨を明記しました。同時に使える人数の章を新設し、回数の上限を環境変数で変えられるようにしました。 iPhone・Safari で文字起こしが失敗する不具合を直しました。音声認識の言語指定を、モデルに応じて正しく送り分けるようにしました。
v0.5	2026 年 8 月	音声モデルを更新（gpt-transcribe / gpt-4o-mini-tts / marin）。

発行：Debate-Class.com / 小林 良裕