CONTENTS

Your setup guide

Everything you need to get started, in one place.

Codex 2

Install Codex and use it with your project.

- Mac setup 3
- Windows setup 4
- Download and set up the project 5
- Use Codex for the session 6
- Common problems and helpful links 7

Vercel 8

Set up Vercel and put your project online.

- Mac setup 9
- Windows setup 10
- Download and set up the project 11
- Test it, then put it live 12
- Update from GitHub 13
- Common problems and helpful links 14

GitHub 15

Download, save and share your project.

- Mac setup 16
- Windows setup 17
- Download an existing project 18
- Create a new project 19
- Save, upload and check changes 20
- Common problems and helpful links 21

Click a title or page number to jump to that section.

STEP-BY-STEP GUIDE

Work with Codex CLI

Install the tools, download a project and make a change you can check.

SET UP
Install + sign in

>

DOWNLOAD
Get project files

>

WORK
Edit + review

In this guide

03 - Mac setup

04 - Windows setup

05 - Download and set up the project

06 - Work with Codex in any project

07 - Common problems and helpful links

Before you begin
You need a Mac or Windows 11 laptop, internet and an OpenAI account that can use Codex. Ask the organiser for the GitHub project link. Install Node.js using the LTS option: it includes npm, the tool used to install Codex. Account limits depend on your plan. Older Windows versions may be less reliable.

How to read this guide
Copy the lines from the dark boxes into Terminal (Mac) or PowerShell (Windows), one at a time. These apps let you give your laptop written instructions. Text labelled CODEX CHAT goes inside Codex instead. Replace example names only when the guide tells you to.

Updated 4 November 2026. Commands checked against the linked help pages. Account sign-in and live uploads were not tested on an attendee account.

MAC / TERMINAL

Install on Mac

Use the Terminal app: press Command + Space, type Terminal, then press Return.

1 Install Node.js

Open the link below. Choose macOS, the LTS version and the .pkg installer. Choose ARM64 for an Apple Silicon Mac or x64 for an Intel Mac (check Apple menu > About This Mac). Install it, then close and reopen Terminal.

Download Node.js (choose LTS)

TERMINAL | run one line at a time

```
node -v
npm -v
npx -v
```

Checkpoint: all three lines print version numbers. If one fails, see page 7.

2 Install Codex

TERMINAL | run one line at a time

```
npm install -g @openai/codex
codex --version
```

3 Create a folder and open Codex

TERMINAL | run one line at a time

```
mkdir -p ~/my-build
cd ~/my-build
codex
```

Choose Sign in with ChatGPT and finish signing in in your browser. Return to Terminal. Read any requests to access your folder before accepting them.

> Ready
> Codex should open and let you type a message. Continue on page 5 to download the session project. If you already have that project, open its folder instead.

WINDOWS / POWERSHELL

Install on Windows

Use Windows 11. Open Start, type PowerShell, and open a normal PowerShell window.

1 Install Node.js

Open the link below. Choose Windows, LTS and the .msi installer (usually x64). Keep npm and Add to PATH selected so the commands can be found. Install it, then close all command windows and reopen PowerShell.

Download Node.js (choose LTS)

POWERSHELL

```
node -v
npm.cmd -v
npx.cmd -v
```

2 Install Codex

POWERSHELL

```
npm.cmd install -g @openai/codex
codex.cmd --version
```

3 Create a folder and open Codex

POWERSHELL

```
New-Item -ItemType Directory -Force ~/my-build
Set-Location ~/my-build
codex.cmd
```

Choose Sign in with ChatGPT and finish signing in in your browser. Follow any Windows setup steps that appear. If your laptop belongs to a school or workplace, you may need their IT team's help.

> Why .cmd?
> Windows sometimes blocks the usual command files. The .cmd versions used here avoid that problem without changing your computer's security settings.

MAC AND WINDOWS / GET THE PROJECT

Download your project

A repository (or repo) is a project saved in Git. Cloning downloads it to your laptop and keeps its link to GitHub.

1 Install or check Git

Mac: run git --version. If missing, run xcode-select --install and finish the installer. Windows: install Git below, keep command-line access selected and reopen PowerShell.

Git for Windows installer

2 Download the organiser's project

Type /exit to leave Codex. Run these lines in Terminal or PowerShell. From the GitHub link, replace OWNER with the account name and REPO with the project name. This creates a folder called my-project.

TERMINAL | run one line at a time

```
cd ~
git clone https://github.com/OWNER/REPO.git my-project
cd my-project
git status
git remote -v
```

For a private project, accept the invitation and sign in first (see the GitHub guide). Use a different folder name if my-project already exists.

3 Add your name and start your work

Use your name and an email linked to GitHub (or the private address in Settings > Emails). The last line creates a branch: a separate place for your changes before they join the shared version.

TERMINAL | run one line at a time

```
git config user.name "Your Name"
git config user.email "YOUR_VERIFIED_EMAIL"
git switch -c my-session-work
```

Check it worked
git status should show my-session-work. git remote -v should show the right GitHub link. Read README for the project setup steps.

MAC AND WINDOWS / USE YOUR PROJECT

Use Codex for the session

Open the project folder you downloaded. You can ask Codex to write code, check it, explain it or fix a problem.

4 Open Codex in the project folder

MAC TERMINAL | Windows PowerShell: use codex.cmd

```
codex
```

5 Understand the project before changing it

CODEX CHAT | paste as one message

```
Read the project setup instructions.
Explain what the main folders are for, what I
need to install, and how to check it works.
Do not change files yet.
```

Follow the project's setup file and the organiser's task. Ask Codex to explain any command you do not understand. Different projects need different tools; only use npm when the project tells you to.

6 Make one change and check it

CODEX CHAT

```
Help me with the task from the organiser.
Start with a short plan, then make one change.
Check that it works and explain what changed.
```

CODEX CHAT | send separately

```
/diff
/review
```

/diff shows what changed. /review asks Codex to look for problems.

> Finish and return later
> Check the changes and follow the project's test instructions. Use the GitHub guide to save and upload your work. Type /exit to leave Codex. To return later, open the same folder and run codex resume on Mac or codex.cmd resume on Windows. Keep passwords and secret keys out of messages and saved files.

HELP / COMMON PROBLEMS

Get unstuck

Start with the exact error message and the terminal where it appeared.

node, npm or npx not found
Install Node.js, then fully close and reopen Terminal or PowerShell and your code editor. Node.js includes npm and npx. If they work there but not in Codex, restart Codex. Ask a mentor if the problem continues.

Mac: installation says EACCES
Your account may not have permission to install this way. Use the npm help link below or ask a mentor. Do not keep adding sudo to commands to try to fix it.

Windows: scripts are disabled / command not recognised
Try npm.cmd, npx.cmd and codex.cmd, as shown on page 4. If they are missing too, reopen the whole command window and check Node.js is installed. Ask a mentor or your IT team if access is blocked.

Cannot sign in or download
Finish signing in in your browser and return to the command window. Check your account can use Codex. If you have reached a usage limit, follow the account message. For download problems, check Wi-Fi and whether it needs a browser sign-in.

Helpful links

Codex CLI: installation, sign-in and first task

CLI commands: review, diff, exit and resume

Codex official repository: npm installation

Windows setup help

Download Node.js

npm installation permission help

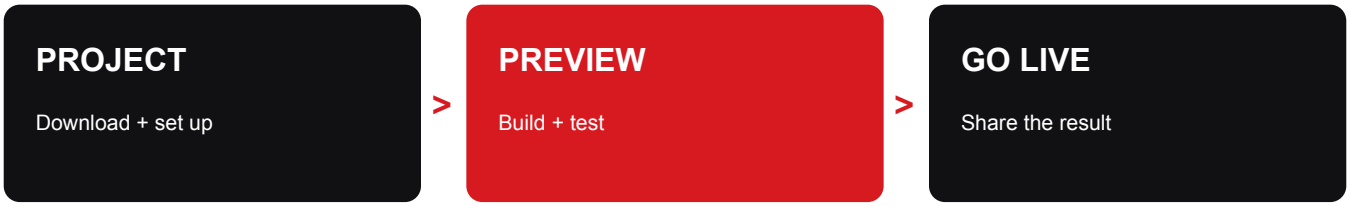
Download a project with Git

You are ready when Codex opens in the right project folder, you understand the setup, and you have checked your changes. Use the GitHub guide to upload your work and the Vercel guide if it needs to go online.

STEP-BY-STEP GUIDE

Set up Vercel

Download a project, test it and put it online when ready.



In this guide

09 - Mac setup and sign-in

10 - Windows setup and sign-in

11 - Download and set up the project

12 - Test and put your project online

13 - Connect GitHub for easier updates

14 - Common problems and helpful links

Before you begin
You need a laptop, internet, a Vercel account and the project link from your organiser. Git downloads the project. Node.js includes npm, which installs the Vercel command tool. Follow the project's own setup instructions and check with the organiser that it can run on Vercel.

How to read this guide
Copy the lines from the dark boxes into Terminal (Mac) or PowerShell (Windows), one at a time. These apps let you give your laptop written instructions. Text labelled CODEX CHAT goes inside Codex instead. Replace example names only when the guide tells you to.

Updated 4 November 2026. Commands checked against the linked help pages. Account sign-in and live uploads were not tested on an attendee account.

MAC / TERMINAL

Set up on Mac

Follow this page, then download and prepare your project on page 11.

1 Install Node.js

Download Node.js for Mac: choose LTS and the .pkg installer for Apple Silicon or Intel. After installing, close and reopen Terminal.

Download Node.js (choose LTS)

MAC TERMINAL

```
node -v
npm -v
```

2 Install and sign in to Vercel

MAC TERMINAL

```
npm install -g vercel
vercel --version
vercel login
```

Follow the sign-in steps shown in the command window. Finish in your browser, then return. Use the Vercel account and team your organiser has asked you to use.

3 Check Git and get the project link

MAC TERMINAL

```
git --version
```

If Git is missing, run xcode-select --install and complete Apple Command Line Tools setup. Ask the organiser for the repository link and any access invitation.

Git installation

> Ready for page 11
> Each version command should show a number, and Vercel should be signed in. Keep using the same Terminal and project folder for the remaining steps.

WINDOWS / POWERSHELL

Set up on Windows

Follow this page, then download and prepare your project on page 11.

1 Install Node.js

Download Node.js for Windows: choose LTS and the .msi installer. Keep npm and Add to PATH selected. After installing, close every command window and open PowerShell again.

Download Node.js (choose LTS)

POWERSHELL

```
node -v
npm.cmd -v
```

2 Install and sign in to Vercel

POWERSHELL

```
npm.cmd install -g vercel
vercel.cmd --version
vercel.cmd login
```

Follow the sign-in steps shown in the command window. Finish in your browser, then return. Use the Vercel account and team your organiser has asked you to use.

3 Check Git and get the project link

POWERSHELL

```
git --version
```

If Git is missing, install Git for Windows and reopen PowerShell. Ask the organiser for the repository link and any access invitation.

Git installation

> Ready for page 11
> Each version command should show a number, and Vercel should be signed in. On the next pages, use vercel.cmd and npm.cmd on Windows if the shorter commands are blocked.

MAC AND WINDOWS / GET STARTED

Download and set up the project

Use the organiser's project. No particular sample app or project type is required.

4 Download the project to your laptop

Replace OWNER with the GitHub account name and REPO with the project name from the link. These lines download it into a folder called my-project. If that folder already exists, use a different name.

TERMINAL | run one line at a time

```
cd ~
git clone https://github.com/OWNER/REPO.git my-project
cd my-project
git status
git remote -v
```

> Private projects
> Accept the GitHub invitation first. The GitHub guide shows how to sign in using gh auth login and gh auth setup-git. Then retry the download. Keep a private project private.

5 Read the setup instructions

README is the project's setup file. Follow its instructions for which tools to install. Use the lines below only if the project uses npm and has a file named package-lock.json. npm ci installs the extra software the project needs.

NPM PROJECT ONLY | Windows may use npm.cmd

```
npm ci
npm run
```

npm run shows the project's available commands. Follow README to choose how to start it, check it or prepare it for upload. If there is no package-lock.json file, or it uses another tool, follow README instead.

6 Check the project on your laptop

If README asks you to copy a settings file and add your own values, do that on your laptop. Do not upload passwords or secret keys to GitHub. Run the project's checks. If the download contains several apps, ask the organiser which folder Vercel should use.

MAC AND WINDOWS / PUT IT ONLINE

Test it, then put it live

Run these commands from the project folder. On Windows, use vercel.cmd if vercel is blocked.

7 Choose the right Vercel project

TERMINAL | run one line at a time

```
vercel link
```

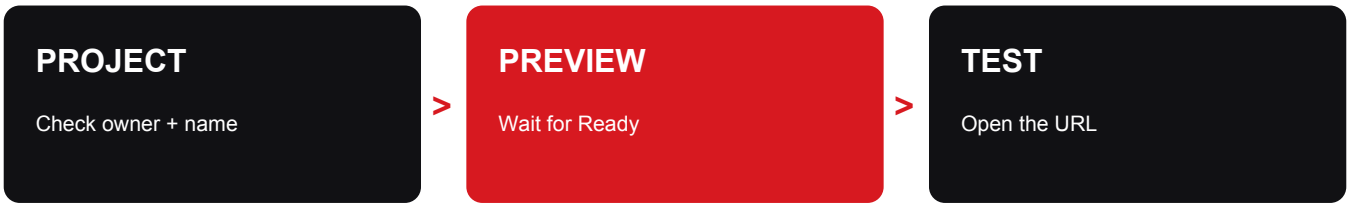
Choose the right Vercel account and team. Select the existing project or create a new one. Check that Vercel is using the correct folder. Follow README for its build settings (the steps that prepare the app to run online). Ask the organiser if you are unsure.

8 Create and test a preview

TERMINAL | run one line at a time

```
vercel deploy
```

Open the test link shown in the command window (the Preview URL). Try the app's main features and check that any connected services work. You may need to sign in to Vercel to open it.



9 Put the project live when ready

TERMINAL | run one line at a time

```
vercel deploy --prod
```

Open the live link shown in the command window or on Vercel (labelled Production). Test it again, then share it. Running this command again replaces the live version with the files currently on your laptop.

> Know what you are uploading
> Check your files before uploading. Keep passwords and secret keys out of anything users can download or see. Add private settings through the Vercel project's Environment Variables page, following README. Upload again after changing those settings.

MAC AND WINDOWS / FUTURE UPDATES

Update from GitHub

Optional: connect GitHub so Vercel updates the project when you upload changes.

1 Upload the project to GitHub

Use the GitHub guide to save and upload your checked files. First check whether this GitHub project is already connected to Vercel. Do not upload node_modules/, .vercel/ or .env files to GitHub.

2 Connect the right GitHub project

In Vercel, open your project > Settings > Git and connect the GitHub project. For a new Vercel project, choose Add New > Project and select it from GitHub. Check the folder and setup settings before continuing.

EXAMPLE GITHUB CONNECTION

Service	GitHub
Project	your-account / my-build
Live version uses	main - verify in settings

Illustrated reference - labels and output may vary by version.

3 Choose how updates go live

Once GitHub is connected, uploads to the branch chosen for the live version (often main) update the live app. Changes on other branches usually get test links. Open Vercel, wait for Ready, and check the link.

EDIT
Change locally

>

UPLOAD
Save to GitHub

>

PREPARE
Vercel updates it

If you upload from the command window
Make and check your changes on your laptop. Run vercel deploy to get a test link. When ready, run vercel deploy --prod to update the live version. Keep using the correct Vercel account and project.

HELP / COMMON PROBLEMS

Fix an upload that failed

Open your project on Vercel and read the upload error message. Start with the first error shown.

npm / vercel missing or PowerShell blocks scripts
Close and reopen the whole command window after installing Node.js or Vercel. On Windows, try npm.cmd and vercel.cmd. On Mac, use the npm help link if installation says EACCES (permission denied).

Wrong project or team
Check the project and team shown in Vercel. From the correct project folder, run vercel link and choose the right project. Make another test upload before updating the live version.

The upload fails or the app does not work
Check the chosen folder and setup settings against README. Try the project's checks on your laptop. Check file names match exactly, including capital letters. Add any required private settings on Vercel and upload again. Check that connected services are working.

Test link asks for sign-in / live version looks old
Sign in to Vercel to open a protected test link. To share the live app, use its Production link. Check that the latest live upload says Ready. Creating a test link alone does not update the live version.

See details about your upload

TERMINAL | replace with the URL from your deployment

```
vercel inspect "https://YOUR-DEPLOYMENT.vercel.app"
```

Helpful links

Install and check the Vercel command tool

Sign in and choose a project

Upload a test or live version

See upload details

Connect GitHub to Vercel

Node.js installers

npm installation help

STEP-BY-STEP GUIDE

Set up Git. Work together.

Download a project, make changes and share them on GitHub.

DOWNLOAD
Get the project

>

SAVE
Save a version

>

UPLOAD
Share + review

In this guide

16 - Mac setup and sign-in

17 - Windows setup and sign-in

18 - Download an existing project

19 - Create a new project if needed

20 - Save, upload and request a check

21 - Common problems and helpful links

Before you begin
You need a laptop, internet and a GitHub account. Git saves a history of changes on your laptop. GitHub stores the shared project online. Install Git and the GitHub command tool (gh). Ask the organiser for the project link and an invitation if it is private.

How to read this guide
Copy the lines from the dark boxes into Terminal (Mac) or PowerShell (Windows), one at a time. These apps let you give your laptop written instructions. Text labelled CODEX CHAT goes inside Codex instead. Replace example names only when the guide tells you to.

Updated 4 November 2026. Commands checked against the linked help pages. Account sign-in and live uploads were not tested on an attendee account.

MAC / TERMINAL

Set up Git and GitHub

Open Terminal with Command + Space > Terminal > Return.

1 Install or check Git

TERMINAL | run one line at a time

```
git --version
```

Finish any Apple installation message that appears. If Git is still missing, run the next line and complete the installer:

TERMINAL | run one line at a time

```
xcode-select --install
```

2 Install the GitHub command tool

If Homebrew is not installed, open brew.sh and follow its installation and Next steps instructions first. Then run:

Homebrew installation

TERMINAL | run one line at a time

```
brew install gh
```

3 Sign in through the browser

TERMINAL | run one line at a time

```
gh --version
gh auth login --hostname github.com --git-protocol https --web
gh auth setup-git
gh auth status
```

Follow the browser sign-in steps and enter any code shown by the command window. Check you are signed in to the right account. Continue to page 18 to download the project.

WINDOWS / POWERSHELL

Set up Git and GitHub

Open a normal PowerShell window from Start.

1 Install or check Git

Install Git for Windows. Keep the option that lets you use Git from the command line and other software. Close every command window, then reopen PowerShell.

Git for Windows installer

POWERSHELL

```
git --version
```

2 Install the GitHub command tool

POWERSHELL

```
winget install --id GitHub.cli --source winget
```

If winget does not work, use the Windows .msi installer from the GitHub link below. After installing, close all command windows and open a new one.

GitHub CLI releases

3 Sign in through the browser

TERMINAL | run one line at a time

```
gh --version
gh auth login --hostname github.com --git-protocol https --web
gh auth setup-git
gh auth status
```

Follow the browser sign-in steps and enter any code shown by the command window. Check you are signed in to the right account. Continue to page 18 to download the project.

MAC AND WINDOWS / GET THE PROJECT

Download and set up a project

Start here if the organiser has given you a GitHub project link.

4 Open the project link and download it

Open the GitHub project link, sign in and accept any invitation. Replace OWNER/REPO below with the account name and project name from the link. The lines below download the project into a new folder.

TERMINAL | run one line at a time

```
cd ~
gh repo clone OWNER/REPO my-project
cd my-project
git status
git remote -v
```

These lines work on Mac and Windows. They create a folder called my-project and remember the GitHub link. Choose a different folder name if my-project already exists.

5 Add your name and email

Use your name and an email linked to GitHub, or the private email address shown in GitHub Settings > Emails. Git uses these to label your saved changes.

TERMINAL | run one line at a time

```
git config user.name "Your Name"
git config user.email "YOUR_VERIFIED_EMAIL"
```

6 Prepare for the session

Open README, the project's setup file. Follow it to install the tools, add any private settings and check the project works. Then run the line below to create a branch: a separate place for your changes before they join the shared version.

TERMINAL | run one line at a time

```
git switch -c my-session-work
```

> Clone, rather than Download ZIP
> Cloning keeps the GitHub link and past changes; Download ZIP only gets the files. If you already downloaded the project with Git, use that folder.

MAC AND WINDOWS / NEW PROJECT ONLY

Create a new GitHub project

Skip this page if you downloaded a project on page 18.

1 Create a folder on your laptop

In Finder (Mac) or File Explorer (Windows), create a folder called my-project in your home folder. Open it in your code editor. Create a file called README.md and write a short project description.

TERMINAL | run one line at a time

```
cd ~/my-project
git init -b main
git config user.name "Your Name"
git config user.email "YOUR_VERIFIED_EMAIL"
```

Replace the example name and email with your own. Git uses them to label your saved changes. These lines work on Mac and Windows.

2 Choose what belongs in Git

Create a file called .gitignore. It tells Git which files to leave out. For a Node/Vercel project, put node_modules/, .vercel/, .env and .env.* on separate lines. Keep passwords and secret keys out of files you upload.

3 Save and upload a private project

TERMINAL | run one line at a time

```
git add README.md .gitignore
git commit -m "Initial project setup"
gh repo create my-project --private --source=. --remote=origin --push
gh repo view --web
```

> Choose who can see your files
> This creates a private project. Only use --public if you want anyone to see the uploaded files. To invite teammates, open the GitHub project > Settings > Collaborators. Choose a different project name if this one already exists.

MAC AND WINDOWS / SHARE CHANGES

Save, upload and check changes

Use the branch you created on page 18. If you skipped that page, run git switch -c my-session-work first.

7 Make a change and check it

Make your change and save the files in your editor. Follow the project's test instructions, then run these lines to see what changed:

TERMINAL | run one line at a time

```
git status
git diff
```

8 Choose the files to save

Replace path/to/changed-file with the file you changed. Repeat git add for other files you want to include. git diff --staged shows the changes you chose. git commit saves them as a named checkpoint on your laptop.

TERMINAL | run one line at a time

```
git add "path/to/changed-file"
git diff --staged
git commit -m "Describe the change"
```

9 Upload your changes and ask for a check

TERMINAL | run one line at a time

```
git push -u origin my-session-work
gh pr create --web
```

git push uploads your saved changes to GitHub. The next line opens a pull request: a request for someone to check your changes before adding them to the shared version. In the browser, check the project and target branch match the organiser's instructions.

> If GitHub will not let you upload
> Ask the organiser for permission to upload. They may ask you to make your own GitHub copy first. Do not force an upload. Wait for the project owner's approval before adding changes to the shared version. Vercel test links appear only when Vercel is connected.

HELP / COMMON PROBLEMS

Get the project working

Check that you are signed in and have opened the right project folder.

git or gh not found / sign-in fails
Close all command windows and reopen them after installing. Run gh auth status to check sign-in. Sign in again if needed, then run gh auth setup-git. Use the browser sign-in steps, not your GitHub password in a Git command.

Repository not found / permission denied
Check the project link, your signed-in account and any invitation. Private projects can look missing if you do not have access. Ask the organiser to add you.

Folder already exists / not a Git repository
Choose a new folder name for the download, or open the project folder you already downloaded. Run git status there to check it works. Do not run git init inside an existing downloaded project or keep trying to create the same GitHub project.

Your upload was rejected
Check which branch you are on with git branch --show-current. Save your work, then ask a mentor to help bring in other people's changes before uploading. Do not use force-push to bypass the problem. If a secret key was uploaded, disable or replace it; .gitignore does not erase past uploads.

Helpful links

Git installation: Mac

Git installation: Windows

Install the GitHub command tool

Browser sign-in

Download a project

Create a GitHub project

Ask someone to check your changes