Three Claude Code projects to build this weekend

Here's how to build the three projects from my video. Each one is harder than the one before, and each one teaches you a different part of Claude Code: a skill, a page, and a task that runs on a schedule. You won't write any code. You tell Claude Code what you want, in plain words.

What you need

- **Claude Code, installed and signed in.** If you haven't done that yet, start with my guide to setting up Claude Code.
- **Project 1** needs nothing more.
- **Project 2** uses an artifact. Artifacts need a Pro, Max, Ultra, Team or Enterprise plan, and a session signed in with `/login`.

- **Project 3** uses a scheduled task. Those are on Pro, Max, Ultra, Team and Enterprise plans too, and they are in research preview, so what they can do may change.

Open Claude Code in a new, empty folder for these projects, for example “Weekend projects” in your Documents.

Project 1 (easy): your own skill

A skill is a file of instructions. Claude Code’s docs say to create one when you keep pasting the same instructions into chat. You can run it by typing `/` and its name, and Claude also uses it on its own when it fits what you ask.

1. Pick one task you do every week. One that goes the same way each time. For example: planning your meals for the week from what’s in your fridge.

2. Say what goes in and what comes out. For the meal plan: the list of what’s in your fridge goes in, and a seven-day plan with a shopping list comes out. Add any rule you always repeat, like how many people you cook for.

3. Ask Claude Code to make the skill. Paste this, with your own task in place of the example:

```
Turn this into a skill I can use every week:
plan my meals for the week from what's in my
fridge. I'll give you the list of what I have.
You give me a seven-day plan and a shopping
list for what's missing. Name the skill meal-
plan. Save it as a personal skill so it works
in every project.
```

The docs say a personal skill lives in a folder with that skill's name inside the skills folder in your home folder's `.claude` folder, in a file called `SKILL.md`. Claude Code creates that file for you. The path is:

```
~/.claude/skills/meal-plan/SKILL.md
```

4. Try it. Type `/meal-plan`, then your fridge list. Claude Code picks up a new skill within the current session. If it doesn't appear in the `/` menu, type `/reload-skills`.

5. Fix it once. Read the result against what you would have done by hand. If something is off, tell Claude Code what to change in the skill. You fix it once, and every later run uses the fix.

Now you never type that prompt again.

Good to know: a skill saved in your home folder is on your machine only. A cloud session, including the cloud scheduled task in Project 3, doesn't read it. The docs say to commit the skill to the repository's skills folder for those.

Project 2 (medium): a subscription finder

Here you hand Claude Code a card statement and get back a page that lists every subscription, what it costs you a year, and what is new or more expensive. The page is an artifact: a live web page that Claude Code publishes from your session to a private address on claude.ai.

Before you start, two things from the docs:

- **Your statement ends up on a claude.ai page.** The artifact is private to you until you share it, but the data on it is on claude.ai. Never press Share on this one. To try it first, use a sample file with made-up rows.

- **An artifact is one self-contained page with no backend.** It can't take a file you drop onto it. To update it, you give Claude Code the new file and ask again.

1. Export your statement. In your bank's app or website, download the last two or three months as a spreadsheet or CSV file. Put the file or files in your project folder.

2. Ask for the page. Paste this:

```
Read the statement files in this folder. Build me an artifact: a page that lists every subscription I pay for, what each costs me per year, and the total. Flag anything that is new or costs more than last month. Keep a small file in this folder with the list you found, so next month you can compare against it.
```

The last line is there because the page doesn't remember anything between sessions. A file in the folder does.

3. Open the page. Claude Code prints the address and your browser opens it. Check it against your statement. If a charge is missing, or something is

listed that isn't a subscription, tell Claude Code and ask it to republish. It publishes to the same address.

4. Every month, update it. Put the new statement in the folder and ask:

```
Here is this month's statement. Update the
subscription page, and flag anything new or
more expensive than last month.
```

If you do this in a new session, give Claude Code the page's address, or run `/artifacts` and attach it. Without one of those, a new session makes a new page instead of updating the old one.

Project 3 (hard): a news briefing agent

Each morning, Claude Code reads the sites and newsletters you follow, picks what matters to you, and gives you a one-page summary. It's the hardest one because it has to run several steps on its own, every day, without you. That takes a scheduled task. The docs describe three kinds, and the choice matters:

	Cloud	Desktop	`/loop`
Runs on	Anthropic's servers	Your computer	Your computer
Needs your computer on	No	Yes	Yes
Needs a session open	No	No	Yes
Sees your local files	No, it starts from a fresh copy of a GitHub repository	Yes	Yes
Shortest interval	1 hour	1 hour in the forms	1 minute

For a morning briefing, a cloud task is the one that keeps going when your laptop is closed. A Desktop task is simpler to start and can read your own files, but only runs while the app is open and your computer is awake. Start with Desktop if you want to see it work today, then move to cloud.

1. Write down what the briefing needs. In a text file in your folder, list the sites and newsletters you follow, what you care about (your industry, your competitors, your customers), and what a good one-page summary looks like to you. Claude Code reads this file on each run.

2. Do one run by hand first. In Claude Code, paste:

```
Read my sources and interests in this folder.
Look at each source for what is new since
yesterday. Pick the five items that matter
most to my business, and write a one-page
summary: one line on why each matters to me,
and a link to each. Keep it to one page.
```

Fix the file and the ask until the summary is one you'd read. A schedule repeats whatever you give it, good or bad.

3. Schedule it. In the Claude desktop app, open the **Code** tab, click **Scheduled** in the sidebar, then **New task**. Give it a name, paste the ask from step 2 as the instructions, pick a schedule such as daily, and choose the time. For a local task, open **Advanced settings** and turn on **Only on this computer**, then pick your project folder. Click **Create**. Or type `/schedule` in Claude Code and describe the task: Claude asks you the follow-up questions and saves it.

Labels in the app have changed names before (cloud scheduled tasks used to be called routines), so look for the same words if a button has moved.

4. For a cloud task, check four things.

- **It starts from a fresh copy of a GitHub repository,** not your computer's files. Put your sources file in that repository.
- **Its internet access is limited by default.** The default environment reaches only an allowlist of package registries, GitHub and similar sites. To read your news sites, add their domains to the environment's allowed domains.
- **The prompt has to stand on its own.** The task runs by itself, so say what to do and what done looks like.
- **Pick a few minutes past the hour,** like 7:07, because a run set exactly on the hour can start several minutes late.

5. Decide where the summary goes. The docs don't say a task emails you. Every run is saved on the task's detail page: click **Run now** to try one, then open the run from the **History** list to read what it wrote. To have the summary sent somewhere, include a connector for a service you already use. The docs' own example posts a summary to Slack. Remove any connector the task doesn't

need, because a run can write through the ones you include.

6. Check it for a week. Read each morning's summary and change the sources file or the ask when something is off.

Good to know: a cloud task acts as you, and its runs count against your usage like any session. Cloud scheduled tasks are in research preview.

Where to start

Do them in order. The skill takes an hour or less, and the subscription page teaches you to hand Claude Code a file. The briefing uses both ideas, and adds the schedule.

Written on 27 November 2026 from Claude Code's own documentation pages on skills, artifacts, scheduled tasks in the cloud, scheduled tasks in the desktop app, and cloud environments. I haven't run these steps myself while writing this guide. Claude changes fast: if a menu has moved, look for the same words in the app.