

Pi replaced every other coding agent harness for me

6 MIN · YOUTUBE · [HTTPS://WWW.YOUTUBE.COM/WATCH?V=JPUN4iLZLDU](https://www.youtube.com/watch?v=JPUN4iLZLDU)

<https://www.youtube.com/watch?v=jPuN4iLZLDU>

SUMMARY

Pi is a minimal coding agent that emphasizes flexibility and user control over its features, making it a unique tool for developers. Unlike other coding agents, Pi allows users to tailor their experience by selecting models and configuring settings to fit their workflow.

- *Pi is designed to be minimal, focusing on user adaptability rather than pre-set configurations.*
- *Installation is straightforward, requiring just a simple curl command.*
- *Users can log in to various models like ChatGPT and Claude, enabling them to select and cycle through models easily.*
- *Key features include the ability to summarize conversations and navigate through previous messages using the `/tree` command.*
- *Extensions can be added to enhance functionality, allowing for custom queries and interactions.*
- *Pi supports a non-interactive mode for terminal use, making it versatile for different coding environments.*
- *The platform encourages user creativity, enabling the development of personalized extensions and workflows.*
- *The community aspect is highlighted, with recommendations for resources and inspiration from other users.*

What's up, y'all? Today, I want to show you Pi, the minimal coding agent harness. It's been my daily driver for a couple weeks now. One thing I've learned, it's not so much about what it has, it's more about what it doesn't have, but what you can do with it. Let's get started and show you what I mean. So, let me start with the base instruction for Codex. So, there's this part where they say, "Run test if you can." Every time I do something in Codex, it starts running test and I don't want it to. Same with Claude code. It has all these instructions and I'm not saying anything of this is bad and for a lot of people, this is probably what you want, but if you want to be of how the agent behaves, `Pi >> >>` is perfect for this. Now, also, you can update some of these settings, but still, you're adapting your workflow to the harness, rather than adapting the harness to your workflow. So, let's get into installing Pi. When you go to `pi.dev`, you get this nice curl command and of course, I'm going to copy paste directly into my editor, get this beautiful animation, install Pi, that's it. It took 6 seconds. Now, out of the box, Pi comes with almost nothing. Let me open it up. No models available. Right now, I can't even use any single prompt. Ping, it's just going to tell me there's nothing selected. So, the first thing I want to do is log in. I'll log in with ChatGPT and Claude. Let's log in. I'll use my subscription and I'll start with my ChatGPT Plus authentication successful.

It's pretty quick. Now, I have GPT 5.5 and I'm also going to log in with my Claude subscription. Okay, authentication successful, too. We can see what models are available by typing `{slash} model`. In here, you'll see GPT 5.5, all the Claude ones, Sonnet, GPT 5 Mini. The first thing I want to do is go to scope models and enable

the ones that I actually use. Fable, the latest Opus at this time of recording. Also, use GPT 5.5 and that's it. Now, press control S to save. Now, if I go to `/scope models`, you'll see I have these three. Why is this important?

Because now I can cycle with control P. Okay? So, Claude Fable 5, Opus 4.8, GPT 5.5. You can also control L to switch through all the models. See, if I press tab here, let's go to Haiku real quick. But, when I do control P again, it goes cycle. So, I'll go with GPT-3.5 and I actually like using a higher thinking level, so you can do that with shift tab. Thinking level high, thinking extra high. I actually use high. Most of Pi's config lives inside of this Pi directory. Show you the settings real quick. Theme is dark, default provider GPT-3.5, thinking level These are the settings. One thing I really like is I can just start with a prompt immediately. Pi, what is the current Pi version?

And we'll open up Pi and immediately run the script. But you can also use non-interactive mode by using `{dash} p`. Because this runs in the terminal, you can type things out. Pi `{dash} p`, explain each file. Here we go. Each of the credentials. There's export. Exports your session as an HTML. You can of course resume any of the sessions. You can share it as well and it will create a secret GitHub gist you can share with your friends. My absolute favorite feature about Pi is `/tree`. And this one took me a little while to understand, but now I got it, okay?

>> >> And I want to show you how it works. I'll ask it, "What are the best ways to get ideas from the audience on YouTube?" Okay, there's a bunch of ideas. Okay, these seem cool. Let's talk a bit more about surveys and forms. Tell me more about surveys and forms. So, now I have this whole conversation and what I can do now is I can say tree and I can go back to any of these messages. I'm going to go to my first message. Now, if I select it, it's going to ask me if I want to summarize the branch. So, if I say no summary, I just jump back and go from there. But if I summarize or summarize with a custom prompt, it will summarize what we talked about. And this is going to be useful because now I'm realizing I want to turn this into a web app. Summarize with prompt, include anything relevant for a feedback app. Now, I can expand the summary. Here's a little bit of the summary. Five questions, multiple choice, avoid asking many questions.

Now, I can clear this. Let's think about creating a new survey app using that stack start. It's actually going to have some of that context, 0.1% of the actual context. So, you could see you could be going back and forth at a glance for a long time. Go back a bit, branch, summarize, >> >> and work your way through a problem. So, instead of you know, copying an entire prompt and copy and pasting an entire inside of a new coding window, you can just do this.

We're not actually going to use any of this, but you can see how it understands the the context from what we've talked about before. I can see `/tree`. You can see the tree here. This is the one we got now. You can label them by shift L and current idea. So, that's tree. Very cool stuff. What about extensions? There's no MCP, no SIP agents, no permission pop-ups, no plan mode. All these things are not included. You can add them yourself if you want to. Now, there's two ways kind of doing things. One is you go to packages here and you search for a package that you think is useful. Pi web access is actually one that I use, but honestly, I probably could write my own that is exactly what I want it to be. Or I can just copy this web access thing, paste.

Now, if I open Pi, you'll see there's a new skill librarian and an extension Pi web access. Now, I can ask it something pretty random that the language model can't know. How many bars are there in Austin? Pi web access extension included these queries. Fast forward, here's this thing, right? This is an example. That's Pi web search. If you need anything else, sub agents, whatever. So, here is where Pi really shines, extending itself. Open up Pi. Here's a very contrived example. Add an extension that I type anything with the string YouTube, the LLM will also say don't forget to like and subscribe.

Here you can see it's reading the docs for extensions. It has some examples, input transforms. Here's another one before agent start. I'm not writing any of this. Pi, the coding agent, is writing this. We can hot reload. Let's reload. Boom, now it's hot reloaded. It will load the extension. I sure love YouTube. Don't forget to like and subscribe. Terrible example, but you get it, right? That you can extend Pi in any way you want. With messages before they go to the agent or after they come to the agent, transform them, log them, anything you can think of. You can probably do it in Pie. I want to give a shout-out to Dylan Meroy who actually got me into Pie with this tweet.

You should check out his pie.files. His is very good. Just if you need some inspiration. Yeah, hopefully this is useful. Let me know in the comments what you thought. I didn't go too deep, but I just wanted to show you kind of what's possible. Yeah, the sky's the limit. All right. Thanks for watching. See you later.