

Claude Code Plan Mode: Think Before You Build

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

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

SUMMARY

Cloud Codes' plan mode enhances the quality of outputs by allowing users to provide context and structure before executing tasks, particularly for complex projects. By using plan mode, users can guide the AI to gather research, ask clarifying questions, and create detailed plans, leading to more effective and concise communication, such as cold email sequences.

- *Plan mode allows the AI to take a step back and strategize before executing tasks.*
- *It improves the quality of outputs by ensuring the AI understands user objectives and context.*
- *Users can create tailored frameworks for cold emails, leading to more effective communication.*
- *Plan mode prompts the AI to ask clarifying questions, filling gaps in user assumptions.*
- *It is session-specific, allowing multiple plans to exist simultaneously in different contexts.*
- *For simple tasks, skipping plan mode can save time and tokens.*
- *The breadth and depth of research conducted in plan mode can significantly reduce the workload.*
- *Users should experiment with plan mode to gauge its effectiveness compared to direct task execution.*

All right, in this video, let's talk about Cloud Codes plan mode. What does it do? Why do we need it? Um, when do we use it? All that good stuff. Um, to give you an example, I created some cold email sequences uh under three different scenarios here. So, uh, it's a little bit cluttered here, but if you can follow along, this is an email example with absolutely no context on how on on how you should write a cold email. So um if you can follow along here um we're using the example of uh a mock context repo for anthropic and we're saying let's use the following files to write a cold email sequence of two emails total pitching clawed code to a CTO at a series A startup right and you can see that I'm giving it some context around company so that would be anthropic in this case competitive positioning and so on but there's nothing about cold email. Right? So, that's the first example. And you can already see where um they're making some uh sort of rookie mistakes like I hope this message finds you well. And look how long just this email is, right? It's really detailed, but no one's really going to read this in 2026.

Um then I have an email example where I have a I have fed it a context of a cold email framework here. But this was this email framework was created with no plan, no plan mode, right? So it was like a oneshot um cold email framework and you can see that it's uh certainly much better than what we had with absolutely no context on how to write good cold outreach emails. But you can still sort of see um where the CTA is like worth a 15-minute look. I can show you how teams are using it. It's it like you might want to end with a CTA at the end, right? to make it to to lessen the cognitive load on the recipient for example, right? But it's not following that because maybe it's not included in the framework, right? Um and then finally, this is an example of an email sequence that is following a cold email framework that was created using plan mode, right? And so you can see that it's uh the shortest. It's probably the most compact and dense in terms of every word being meaningful and

actually uh serving some sort of value. And the CTA is uh super clean, right? As per the framework that that we fed it. And so, um, a caveat here is the LLM is only going to be as good as the context you feed it. And if this is not the style in which you like to write your cold emails, then no problem. You just have to uh create a new framework, right, that mimics how you like your cold emails to sound. But the point here is that um this email example that follow the framework that was created using plan mode is pro is the most in line and aligned with what is considered to be best practice in 2026.

Okay. Um so that's what we're seeing in this um in this screen. And so hopefully this gives you an idea of the the increase in the capability of an LLM and the quality of its output and outcome that you can expect when you use plan mode. So what is plan mode? How do you use it? Um let's get right into it. Right. Actually I'll uh start with the with this. Right. So plan mode comes on uh after I type it in. You can also just ask Claude. Uh you can just say, "Can you go into plan mode?" And sometimes uh Claude will volunteer to go into plan mode if the task you're giving it is like really verbose and really lengthy and really complex. But I might say something like this, right?

Um, hey, so we have a competitive positioning markdown file, but I want to go really deep against our main competitor uh, which is open AAI, right? and they have their own um uh cloud code equivalent or a uh CLI um that is specifically designed for coding and it's called uh codec. Can you um create a plan to thoroughly research Codeex uh and highlight its competitive positioning, our competitive positioning against it, pros and cons, how people view the two competing products, and how we might be able to pitch cloud code as being superior or outline a clear set of tricks.

trade-offs for people to make easier decisions on when they should opt for cloud code over codeex. Um, while it's doing this, plan mode really gives sort of permission to cloud code to take a step back and think about what you just told it and map out like a series of steps that you should probably take. And it also gathers a lot of research ahead of time, right? And you might see a similar pattern when you start creating a larger scale context repo. For example, it might go and start reading a lot of existing content. And you can see that here already. It explored existing competitive positioning content. So it probably went and read the uh competitive positioning doc. Right.

Okay. So, it's written a preliminary plan here. And it the other thing that I really like about plan mode is often times it'll ask a bunch of questions to make sure that um to to understand your objectives, what how you're thinking, right? Often times when you give instructions to Claude, there's going to be gaps where you're assuming a lot of things and not verbalizing them.

And in plan mode, Claude is more likely to ask you these questions to explicitly align on some of these assumptions that you're making uh subconsciously, right? And so where do I want to save them? I will save them directly in context. Um I will say cover all equally and I will submit and I'll submit answers. So you can encourage Claude to to ask more questions because often times it's almost easier to um have them ask you questions versus you trying to uh address all the edge cases and uh verbalize all the assumptions that you're making. Right?

So Claude says yes there are some assumptions that I'm making. So um who is the primary audience for this battle card? Uh let's say it's internal GTM. What competitive positioning stance should we take? Complimentary head-to-head defensive protect share. We'll say head-to-head. How should we handle the open source gap? Um we should address proactively why cloud code is close source. Should the document include a competitive intelligence monitoring section?

No. Don't keep it static. Submit answers. Right? So, there were a bunch of things that Claude was just going to go ahead with that it stopped itself and said, I should probably check with the user to see if these are the right decisions. Right? So, now it's created a plan to create this battle card document. As you can see here, uh implementation steps, step one, create battle card document. So, this plan is pretty thorough, right? And it actually includes a lot of um the research that it's done already. And you are given a few options. You can just clear context and auto accept edits. You can auto accept edits without clearing context or you can manually approve edits. Usually the I find that the plans are quite thorough. And so I'll go through them to make sure that there's enough um context in there. And if yes, then I'll just clear the context because then it really frees up in case you have follow-up tasks that come out of the uh whatever task you're doing, right? You want to preserve that context window uh as much as possible. So I will go ahead and clear context and it'll just dump the plan in here and it'll start creating the document. Right.

While it's doing this, um, I want to highlight a few other features of, uh, the plan mode. And so, um, I'm opening a new window here. I'm just going to cla start a new session. You will see that, um, each claude session has its own plan. So for example, if I say um if I do plan right uh to go into plan mode and what you can do um in plan mode is actually type in slash plan again and what it does is if you do if you uh type in slash plan while you're while plan mode is on or while you're already in plan mode it shows you the plan but in this case it doesn't because uh this is a new session and So this is not affected by this plan, right? So uh it's good to know that um a plan is at a session level. It's not at a repo level.

It's not a project level. It's not a a computer level. It's at a session level. So you can have multiple plans going at the same time uh in the same context repo or in any repo. Um, the other thing that I'll mention is if you're doing something really simple like um let's tighten up the wording, right? Uh, of this file and I know exactly what I want to change. Don't go into plan mode, right? Cuz it's um it eats up a lot of tokens. It uh takes a lot of time, right? So sometimes that trade-off is not worth it. And so if you're doing something really quick uh and simple, just uh skip the plan mode and just go directly to the task. If you're doing something complex and you'll start to get a good feel for it because as you start seeing sort of the gap in the quality of the output um as you use plan mode and as you don't as you skip plan mode, you'll start to get a good feel for it. Um, and sometimes COD will prompt you to go into plan mode to let it go into plan mode because it feels that it's warranted for the complexity of the task that you're trying to do.

All right. So, the battle card document has been created here and uh per the plan and here's a summary of what the content looks like. If we just go and kind of take a look real quick, you can see um that it's for internal it's for an internal audience. So we have like a 60cond TLDDR um what codec CLI might claim and how to reframe their claims. uh a little bit of landscape overview API pricing which is really helpful but as with everything I

think you you always have to double check their work right whether you use plan mode or not right um because sometimes it's just not going to be accurate I think like for example output token for opus 4.5 I think it's come down to something like \$25. Um, so yeah, there are some bits and pieces here where you'll probably do another round of back and forths with claw to make sure that everything is accurate, but just the the breadth and the depth of research that it's done using plan mode um is pretty impressive, right? And this is probably 80% of the work um that's already done. And you just need to go in and and and tighten up um each of these based on if you're working anthropic based on what you've seen in the field, right? In terms of your um what you know uh the context that you have in your brain, uh you would you would give that to uh Claude to further refine this. Um so yeah, this is uh a quick this has been a quick overview of of plan mode. I hope that um you know you're able to take away a few things. I think the TLDDR really is whenever you feel like you're doing something that's a little bit complex and you don't necessarily want Claw to jump straight into doing that task or executing that task, but you wanted to sort of take a step back and plan out a sequence of things that it needs to do and maybe gather some uh intel or research uh initially, uh then always use plan mode. And um as with everything, just uh experiment using plan mode and checking the quality of the output versus when you don't use plan mode. And I think pretty quickly you'll start to get a feel for when to use plan mode versus not. Cool. Um as with um as always, just uh feel free to reach out if you have any questions. Um and hopefully you found this helpful.

Thanks.