MY WEIRD PROMPTS

Podcast Transcript

EPISODE #103

The Future of Coding: Is Your Brain Wired for AI?

Published December 25, 2025 • Runtime: 23:44

https://myweirdprompts.com/episode/cognitive-fit-programming-ai/

EPISODE SYNOPSIS

In this episode of My Weird Prompts, Herman and Corn dive into a thought-provoking idea from their housemate Daniel: the redefinition of the "developer" in the age of artificial intelligence. As we reach the end of 2025, the duo discusses why traditional coding hierarchies are crumbling as AI takes over the burden of syntax, shifting the human focus toward architectural oversight and "cognitive fit." From the frustrations of JSON to the tactile nature of Docker, they explore how different brains process logic and why a 20-language experiment might be the future of tech education. Learn why you might have a "SQL brain" and how AI is acting as the ultimate translator between human intuition and machine execution.

DANIEL'S PROMPT

Daniel

I'd like to discuss the evolving role of developers in the age of AI. While there's a constant push for more developers, I think we need to redefine that role to include people who are tech-savvy and understand the development process at an execution level using AI tools. There's a huge variety of languages and formats like YAML, JSON, SQL, Python, JavaScript, HTML, and more. In my experience, what's considered 'easy,' like Python or JSON, isn't always intuitive, while tasks like DevOps or working with Docker can feel simpler. I believe people's brains are wired differently, and not everyone is cognitively suited for every type of coding. I think it would be a fascinating experiment to take someone with no coding experience and give them elementary tasks across many different languages to see which ones they naturally excel at. Has anything like this been tried? What do you think of this idea, and how could we map individual cognitive styles to the programming languages they are most naturally suited for?

TRANSCRIPT

Corn

Hey everyone, welcome back to My Weird Prompts! I am Corn, and I am feeling especially relaxed today. I guess that is just the sloth life for you, but I have been looking forward to this conversation all week.

Herman

And I am Herman Poppleberry. It is great to be here, and it is especially great to be discussing this today, December twenty-fifth, twenty twenty-five. Happy holidays to everyone listening. We have a really fascinating prompt today from our housemate, Daniel. He has been thinking a lot about the changing landscape of technology, and specifically, what it means to be a developer in this age of artificial intelligence.

Corn

Yeah, Daniel always has these deep thoughts while we are hanging out in the kitchen in Jerusalem. He was telling me that the whole definition of a developer is shifting. It is not just about someone sitting in a dark room typing away at a green-on-black screen anymore. It is becoming something much broader.

Herman

Exactly. The prompt explores this idea that as artificial intelligence tools become more integrated into our daily work, we need to redefine who a developer actually is. Daniel suggests that we need more people who are tech-savvy and understand the development process at an execution level, even if they are not traditional computer science majors. They are using these new tools to bridge the gap between an idea and a finished product.

Corn

It is like the bar for entry is lowering, but the complexity of what you can build is actually going up. But the thing that really caught my ear was Daniel talking about how different languages feel to different people. He mentioned that even things that are supposed to be easy, like Python or JSON, can feel really unintuitive, while something like Docker or DevOps might feel simpler to certain brains.

Herman

That is such a profound observation. We often talk about programming languages as if they are a ladder of difficulty, with HTML at the bottom and C-plus-plus or assembly at the top. But Daniel is pointing out that it might be more about cognitive fit than objective difficulty. It is about how your brain is wired to process logic, hierarchy, and structure.

Corn

I love that idea. It makes me feel better about the times I have tried to look at code and just felt my brain turn into mush. So, Herman, let's start there. In this year, twenty twenty-five, how much has the role of a developer actually changed compared to, say, two years ago?

Herman

It has changed massively, Corn. Two years ago, a developer spent a huge portion of their time on syntax. They were worrying about where the semicolons go, or if they used the right library call. Today, in twenty twenty-five, we have these incredibly sophisticated artificial intelligence agents that handle most of that boilerplate. A modern developer is more like a conductor of an orchestra. You need to know what each instrument does and how they should sound together, but you are not necessarily playing every single violin part yourself.

Corn

So the skill is shifting from writing the code to knowing what to ask the AI to do?

Herman

Partially, yes. But it is also about verification and architecture. You have to be able to look at what the AI produced and say, wait, that is going to create a security vulnerability, or that is not going to scale when we have ten thousand users. You need to understand the development process at a high level. That is why Daniel's point about being tech-savvy is so important. You need to understand the flow of data, even if you are not the one writing the specific functions.

Corn

That makes sense. But let's talk about those languages Daniel mentioned. He said JSON drives him crazy. For the people listening who might not know, what exactly is JSON, and why would it be frustrating?

Herman

JSON stands for JavaScript Object Notation. It is basically a way to format data so that computers can easily read and write it. It uses a lot of curly braces, square brackets, and quotes. If you miss one comma or one bracket, the whole thing breaks. For some people, that level of strict, nested hierarchy is very clear. They see the structure instantly. But for others, it is just a sea of punctuation that makes it hard to see the actual information.

Corn

I can see that. It is like looking at a map where all the roads are the same color. And what about Python? Everyone always says Python is the easiest language to learn. Why would someone find that hard?

Herman

Python is designed to be readable, almost like English. But it relies heavily on indentation to define blocks of code. If your brain doesn't naturally track white space as a structural element, it can be very confusing. Also, Python abstracts a lot of things. It does a lot of work behind the scenes. For a certain type of brain, that abstraction is frustrating. They want to see exactly how the engine is working, not just have the car drive them there.

Corn

That is interesting. So when Daniel says he finds DevOps or Docker easier, what does that say about his brain?

Herman

Docker is all about containers. It is very procedural and environmental. You are saying, I want a box, I want this operating system inside it, I want these three tools installed, and I want it to talk to this other box. It is very tactile in a digital sense. It is about building a system, not just writing logic. Someone who excels at that might be more of a systems thinker. They see the big picture and how different parts interact, rather than focusing on the minute details of a specific algorithm.

Corn

I think I am a systems thinker. I mean, I am a sloth, so I am very good at understanding how a branch connects to a tree and how that tree fits into the forest. I just take a long time to move between them. But I really like this idea that we are all wired differently for different types of technical work.

Herman

It is a complete shift in how we think about education and hiring. We have spent decades trying to push everyone through the same coding bootcamps, teaching them the same five languages. But what if we are forcing people into roles that don't fit their cognitive architecture?

Corn

That brings us to Daniel's big idea. He suggested an experiment where you take someone with zero experience and give them tiny tasks in ten or twenty different languages to see which ones they naturally gravitate toward. Has anyone actually tried that?

Herman

Not in a widespread, systematic way, but there is some historical context for it. Back in the nineteen sixties and seventies, companies like IBM used things like the Programmer Aptitude Test. It was more about logic puzzles and pattern recognition. But those were designed for a world where you had to be a mathematical genius to code. We are in a different world now.

Corn

Right, because now the AI can handle the math. We need to find the people who can handle the logic and the structure. Let's dig deeper into what that experiment would look like, but first, I think we need to hear from our friends who keep the lights on around here.

Herman

Good idea. Let's take a quick break. Larry: Are you tired of your dreams being limited by the physical constraints of your own skull? Introducing the Dream-Catcher Five Thousand. This revolutionary head-strap uses bio-resonant copper coils to broadcast your subconscious thoughts directly into a cloud-based storage locker. Never forget a million-dollar idea again! Simply strap it on before bed, and wake up to a downloadable PDF of your nightly hallucinations. Warning: Side effects may include vivid recurring dreams about being a middle-manager in a paperclip factory, temporary loss of the ability to recognize your own reflection, and a persistent smell of burnt toast. The Dream-Catcher Five Thousand - because your brain is too small for all your big ideas! BUY NOW!

Corn

Thanks, Larry. I think I will stick to my regular dreams about hibiscus flowers, but that sounds... intense. Anyway, Herman, back to this experiment. If we were to set up a test for twenty different languages for a total beginner, what would that even look like?

Herman

It is a brilliant concept. You would want to strip away the intimidating setup process. No installing compilers or managing paths. You would use a web-based interface where the environment is already set up. Then, you give them a very simple goal, like Daniel said, a print statement. But you do it across different paradigms.

Corn

Give me some examples. What would the different paradigms be?

Herman

Okay, so you start with a functional language like Haskell. It is very mathematical and pure. Then you move to a procedural language like C. Then an object-oriented one like Java. Then you move to data-driven formats like SQL or JSON. And you also include what we call declarative languages, like CSS for styling or YAML for configuration.

Corn

And you just watch how long it takes them to get it?

Herman

Not just the time, but the frustration level and the "aha" moment. You could use biometric sensors to track their stress levels. Does their heart rate spike when they see a nested loop in Python, but drop when they are writing an SQL query? You could also look at how they troubleshoot. When the code doesn't work, where do they look first? Do they look at the syntax, or do they look at the logic flow?

Corn

I bet you would find people who are absolute naturals at SQL but can't wrap their heads around JavaScript.

Herman

Exactly! SQL is a great example. It is basically asking a database a question in a very specific way. Select this column from this table where this condition is true. It is very linguistic. If your brain is wired for language and categorization, SQL feels like magic. But if you try to teach that same person how to manage memory in C-plus-plus, they might feel like they are hit by a wall.

Corn

It feels like we are talking about a new kind of career counseling. Instead of just saying "go into tech," we could say "you have a YAML brain" or "you have a JavaScript brain."

Herman

That is exactly the goal. And in twenty twenty-five, this is more relevant than ever because the AI can act as the translator. If I am a brilliant system architect but I hate writing Python, I can use an AI to generate the Python code based on my architectural requirements. The AI fills in the gaps where my cognitive style doesn't naturally align with the language.

Corn

So, we are moving toward a world where the specific language matters less, but the way you think matters more. But wait, if the AI can translate everything, does the language even matter at all?

Herman

That is a great question. I think the language still matters because it provides the framework for how you interact with the computer. Even with AI, you are still operating within the constraints of what that language can do. If you are building a website, you are still dealing with the DOM and CSS, even if an AI is writing the code. You still need to understand the underlying structure to tell the AI if it did a good job.

Corn

It is like being a chef. You might have a robot that can chop the onions and stir the sauce perfectly, but you still need to know how those flavors work together to create a great meal. You need to know that the onions need to be sauteed before the tomatoes go in.

Herman

Perfect analogy, Corn! And some chefs are great at baking, which is very precise and scientific, while others are great at making stews, which is more about intuition and slow development. Neither is better, they just require different mentalities.

Corn

I would be a great slow-cooker chef. Low and slow, that is my motto. But let's go back to the experiment. If we were to map these cognitive styles to languages, what are some of the categories we might find?

Herman

Well, based on current research in cognitive science and developer productivity in twenty twenty-five, we could probably identify a few key styles. First, you have the Hierarchical Thinkers. These people love things like JSON and XML. They see the world in nested trees. They are great at organizing complex data structures.

Corn

Okay, who else?

Herman

Then you have the Procedural Thinkers. They like a clear step-by-step flow. Do this, then do that, if this happens, go here. They tend to do well with languages like C or Go, or with DevOps tools like Docker where you are defining a sequence of events.

Corn

I think I might be a procedural thinker. I like a clear path. What about the people who like the "easy" languages?

Herman

We might call them Abstractionists. They don't want to worry about the details of how the computer works. They want to work with high-level concepts. They love Python and Ruby. They want to say "sort this list" and not worry about whether it is a bubble sort or a quick sort.

Corn

And then there are the Visual-Spatial Thinkers?

Herman

Yes! They excel at CSS and front-end development. They can visualize how a change in a line of code will shift an element on the screen three hundred pixels to the right. To them, the code is a physical object they are moving around.

Corn

This is so cool. It makes the tech world feel so much more inclusive. It is not just for the "math people" anymore. It is for the organizers, the builders, the visionaries, and even the sloths who just want to make things work better.

Herman

It really does. And think about the implications for the global economy. We are always hearing about the "developer shortage." But maybe there isn't a shortage of talent, just a shortage of people who fit the traditional, narrow mold of what we think a developer should be. If we can identify people's natural cognitive strengths early on, we could bring millions of talented people into the tech workforce who previously thought they "weren't good at coding."

Corn

Daniel mentioned that he finds Python hard but Docker easy. That is such a specific example of this. Docker is often considered "advanced" or "harder" because it involves servers and infrastructure. But for Daniel, the logic of it is clearer.

Herman

Right. To him, Python might feel like he is trying to write a novel in a language where the grammar rules keep changing based on how many spaces he uses. But Docker feels like he is building a Lego set. It doesn't matter how many pieces there are, as long as the instructions make sense and the pieces fit together.

Corn

So, if someone is listening to this and they want to get into tech, but they have been intimidated by Python or JavaScript, what should they do?

Herman

My advice would be to try exactly what Daniel suggested. Don't just stick with one language because a website told you it was the easiest. Spend an hour with five different things. Try writing a simple SQL query. Try styling a basic web page with CSS. Try setting up a simple automation with a tool like Zapier or a YAML configuration. See which one makes your brain feel energized rather than exhausted.

Corn

That is great advice. It is like trying on different pairs of shoes. You don't know which one fits until you walk around in them for a bit.

Herman

And don't be afraid to use the AI tools we have in twenty twenty-five to help you. If you find a language you love but you struggle with the syntax, let the AI handle the syntax while you focus on the part you are good at. That is what being a developer is now. It is a partnership between human creativity and machine execution.

Corn

I love that. A partnership. It makes the future feel a lot less scary.

Herman

It really does. We are moving away from the era of the "code monkey" and into the era of the "tech architect." It is a much more human-centric way of working with computers.

Corn

Well, this has been a really eye-opening conversation. I feel like I understand my own brain a little bit better now, even if it does move at a sloth's pace.

Herman

Your brain is perfect just the way it is, Corn. It provides a very important perspective on this show.

Corn

Thanks, Herman. And thanks to Daniel for sending in such a thought-provoking prompt. It is amazing how a simple question about JSON can lead to a whole discussion about the future of human cognition.

Herman

That is the beauty of My Weird Prompts. We never know where these ideas will take us.

Corn

Before we go, I want to remind everyone that you can find us on Spotify and at our website, myweirdprompts.com. We have an RSS feed for subscribers, and there is a contact form if you want to send us your own weird prompts. We would love to hear from you.

Herman

Yes, please do get in touch. We love exploring these topics with you all. And remember, whether you have a Python brain, a SQL brain, or a Docker brain, there is a place for you in the future we are building.

Corn

Well said, Herman. This has been My Weird Prompts. Thanks for listening, and we will talk to you next time!

Herman

Goodbye everyone!

Corn

Bye! Larry: Still listening? Why? You should be out there buying a Dream-Catcher Five Thousand! Your subconscious is full of untapped data points just waiting to be monetized. Don't let your nightly visions go to waste. BUY NOW!