

**Lex Fridman Podcast #461 - ThePrimeagen: Programming, AI, ADHD, Productivity,
Addiction, and God**

Published - March 22, 2025

Transcribed by - thepodtranscripts.com

Lex Fridman

The following is a conversation with Michael Paulson, better known online as ThePrimeagen. He is a programmer who has entertained and inspired millions of people to have fun building stuff with software, whether you're a newbie or a seasoned developer who has been battling it out in the software engineering trenches for decades. In short, ThePrimeagen is a legendary programmer and a great human being with an inspiring roller coaster of a life story. This is the Lex Fridman Podcast. To support it, please check out our sponsors in the description. And now, dear friends, here's ThePrimeagen. What do you love most about programming? What brings you joy when you program?

ThePrimeagen

I can tell you the first time that I ever felt love in programming, or felt that joy or that excitement -

Lex Fridman

Sure.

ThePrimeagen

Which was in college. It was the second class, data structures, and the teacher that was teaching Ray Babcock, he was talking about linked lists. Now you have to learn Java at Montana State University when I went, and so he's off there explaining this whole linked list thing and all that, and then he shows code. And in the code it's abstract class node or whatever it was, I can't remember what it was. And then it had a private member, and that private member was of type node, and I've never seen that before. It is a class that is called node with a member that is of itself. And for the first time ever I was like, "Oh my gosh. There's no end. There's no way to iterate. This is not a set of 10 items. This is a set of infinite items." And so my mind kind of exploded in that moment, like, "What you can express is huge. I can see what memory looks like. I can see this hopping through space." And I just remember being just so blown away, because up until that point, everything was just, "All right, I have a list of 10 items. I have a list of 20 items." Right? It was very rigid and small, and the things I built were really small and trivial, and all of a sudden I felt like I could build anything in that one moment. And it was so amazing. I just remember sitting in class for, I don't even remember how long those classes were or anything, but I just remember being just completely profoundly impacted by this notion. And so I just sat there and I watched, and I had the exact same experience in heaven's forbid by a software engineering class, when we talked about the decorator pattern, where you can keep on constructing these objects in this recursive way. Not that I think that's actually a good idea to do, but just watching that and realizing there's so many weird and unique ways you can solve problems, and anything your mind can think of, you can just create that. And I just remember getting just so excited about the possibility that anything is possible.

Lex Fridman

Yeah, let's wax philosophical about a linked list. It is pretty profound. For people who don't know, a node in a linked list doesn't know anything about the world it's in. It only knows about the thing it's linked to, its neighbor. Maybe that's symbolic. It's a metaphor for all of us humans. There's billions of us on this planet and we only know about our local little network.

ThePrimeagen

Yeah.

Lex Fridman

And it's kind of beautiful. And you realize in that little simple data structure, you can construct arbitrarily large systems, and they're like roots that go through memory. And then of course, that's where you get all the programming languages that allow you to dump junk into memory and have memory leaks, and therefore create infinite pain as you try to figure out where that unfreed memory is. For me, yeah, probably... It's so beautiful the way you put that. Linked lists are indeed beautiful. Recursion also for me, when I finally wrapped my brain around what it means to write a recursive function.

ThePrimeagen

What was the thing? What was one that taught you? Because I think we all... You probably did factorial, where you just do a quick factorial of it. It just doesn't hit home. What was the thing that made it hit home?

Lex Fridman

I don't remember the first.

ThePrimeagen

I remember my first. How do you not remember your first? It was magic.

Lex Fridman

I've had so many that it just...

ThePrimeagen

I mean, you are a Lisp guy. You're probably pretty used to the recursion.

Lex Fridman

Yeah, all I remember is just surrounded by sea of parentheses. I mean that's really, probably, when I... In high school, I think it was either Java or C++. Wow, how do I not remember that? It must have been C++. And then college, the generic bullshit software engineering classes were Java, but then the renegades, the cool kids, were all using Lisp. That's when you're doing the AI, the quote-unquote "AI" at that time, that that was Lisp. If you want to write a chess engine, you would use Lisp. And so for me, probably the moment I really fell in love

with programming was Lisp, and writing Othello programs and chess engines, all kinds of engines that play a game, and then I could play against that thing and that thing would beat me. The joy of being destroyed by the thing you've created. And oh, game of life too. Cellular automata. That's when I... I built that, you know, all kinds of programming languages. That's less about programming languages and more about the system you create. And that just filled me with infinite joy, having... Now similar to the linked list situation, creating a system where each individual cell only knows about its neighbors and operates in very simple rules. But when you take that system as a whole and allow it to evolve over time, you can create infinite complexity. So I just... Man, those are many pothead moments, where I'm just looking at the beautiful complexity that can be created with cellular automata. That filled me with just infinite joy, for sure. But yeah, all I remember is parentheses. So my memories of my first are drowned in a sea of parentheses.

ThePrimeagen

Oh man. Well, first off, mine was in Java, so my first was a little bit more rigid, kind of, you know, a corporate experience.

Lex Fridman

Yeah. Cold, meaningless...

ThePrimeagen

But... yeah. I was in a lab, everyone was using CentOS at that... or Cent OS or however you say. I always called it CentOS, the fresh maker.

Lex Fridman

Yeah.

ThePrimeagen

And so it's just like I'm in this very cold...

Lex Fridman

That's nice.

ThePrimeagen

Thank you.

Lex Fridman

Yeah.

ThePrimeagen

I'm in this cold, rigid environment with my Microsoft keyboard, programming away in Java. And I still have just such... this memory of despair, because I love programming, this was

after the linked list, and I cannot figure out recursion. And so I go to the university store and I buy a book and it's Deitel and Deitel learn Java and it has a section, Recursion, so I open it up and I start reading it, and it just doesn't hit home. And I'm spiraling into this.

Lex Fridman

Yeah.

ThePrimeagen

Like, "Maybe I'm not a programmer. Maybe I'm not worthy enough to enter into this circle of people who can figure out what the heck recursion means." And Deitel and Deitel's, I still remember this, their exact phrase was, "Every young budding developer solves this recursion program," and it was the Tower of Hanoi. And guess what? I don't know if I can solve the Tower of Hanoi to this day. It's a very hard recursive problem. And I just sat there and thought, "Oh my gosh. I'm not going to make it." And I sat there in the lab for eight hours, 10 hours doing these things, so worried. It's the week of recursion, we have to do a lab assignment. "I'm not going to be able to do it." And I just remember being genuinely worried about that. And then... Because always my big problem was like, "Okay, do factorial." Why not just use a for loop? "Okay, what about Fibonacci sequence?" Why not use a for loop? I don't understand. What's the purpose of recursion? I don't understand it yet. It's so powerful. Why? It looks like a really complicated for loop. And so I just could not understand it. And then lab came that day and it was, "I'm going to give you a 2D array you have to read from a file. This is what a starting position looks like. This is what an ending position looks like. This is what a wall looks like. I want you to find me a path through the maze." So I just sat there like, "Okay, well I guess I can just go up and I can create a visited grid, so I know not to visit these places anymore." And then all of a sudden it just started clicking. Like, "Well, wait a second. I don't know the maze, but if I just go up, right, down, and left, and hop back every time I've been to that square, don't visit it, it will just go forever." And I realized in that moment, I'm like, "I actually understand. I've understood recursion this whole time, I just never had a problem in which it actually made sense to use. And that was my big downfall, is that I was measuring my understanding with the problems that I had available, which were just list traversal, which is not a good use of recursion. And so I just remember that freeing... Oh, man. Recursion. It was a great moment in my life.

Lex Fridman

I mean it does require, to be fair, a leap of faith, because people will tell you, those conformist, dogmatic, Java instructors will tell you, that this is important, to understand recursion. But it takes a leap of faith that this is a different way of looking at the world, and it's a powerful way of looking at the world. Actually, I think I remember my first now.

ThePrimeagen

All right.

Lex Fridman

I think it was [inaudible 00:09:42] first search for one of the games. Maybe Othello, something like that, and for that implementing recursion. Understand that you can search trajectories through the space of states and do that recursively? That was mind blowing.

ThePrimeagen

Yeah.

Lex Fridman

Just imagining the possibilities.

ThePrimeagen

You can just see it all. Yeah.

Lex Fridman

Yeah, just numbers flying. It was like The Beautiful Mind. And that's when I also discovered conspiracy theories, and I just saw. I saw the truth. Okay, yeah. So what were we talking about? Oh, what was the most painful aspect of programming for you? What memories do you have of deep, profound suffering in terms of programming in the early days?

ThePrimeagen

I would say the biggest one that I can really hold on to had to be one of two experiences. The first experience was when I was at a place called Schedulicity, and... Am I not allowed to say the place [inaudible 00:10:43]

Lex Fridman

You're allowed.

ThePrimeagen

I'm not sure if they're even operating still at this point, but they're in-

Lex Fridman

There was something funny about the name. I'm sorry.

ThePrimeagen

Oh, Schedulicity? Yeah. Actually, the name was so bad that when you looked at their paid-for Google ad terms that they would make sure that they're at the top of the list, the spellings were just insane, because no one knew how to spell the word Schedulicity, and so the Google optimizing for that is just hilarious. But okay, go back to the thing. The thing that kills me the most about programming, what I actually considered the worst aspect of programming, is when you know everything. And so when I was at this job, it's just every single day I'd come in, there were no surprises, there was no questions. I didn't understand

the code base, sure, that's fair. I didn't understand all the things about the code base. But I knew I was going to go in, I was going to generate some sort of object from the database. I was going to take that object from the database, and I was just going to map it over and just display it on the webpage. There's no creativity, there's nothing to it. It's very almost factory line kind of work. And that was a very difficult moment for me, which is... I didn't enjoy programming, because I knew everything about it. I already knew exactly what I was going to do that day. I knew all the hurdles I going to have to go over. There was no unknown unknowns, if you will. It was just knowns at all times. And for me, that is the worst part about programming, is when you already know the solution and it's just a matter of how fast you can type and get it out from your head to your hands.

Lex Fridman

So the absence of uncertainty, the absence of challenge, was the pain?

ThePrimeagen

Yeah.

Lex Fridman

That's pretty profound, Prime.

ThePrimeagen

I'm more than just good looks. I want you to know that.

Lex Fridman

It's a low bar. What do you identify as? I'm enjoying asking the general question.

ThePrimeagen

38, male.

Lex Fridman

Male.

ThePrimeagen

Husband of beautiful wife.

Lex Fridman

Okay. You stream about all kinds of programming, but what kind of programmer are you? Are you full-stack developer, web programming? And maybe can you lay out all the different kinds of programming and then place yourself in that, in terms of your identity. Sexual identity as well.

ThePrimeagen

Yeah, we can put it all in there.

Lex Fridman

Okay.

ThePrimeagen

Plus, obviously those two are very, very tightly coupled.

Lex Fridman

I have seen you on the border of sexually aroused by certain languages. I think you got real excited about OCaml, or...

ThePrimeagen

OCaml. Let's go. Thank you Dillon Mulroy [inaudible 00:13:08]

Lex Fridman

Okay, wow.

ThePrimeagen

Yeah.

Lex Fridman

I did not expect that. That escalated quickly. Anyway, what do you identify as?

ThePrimeagen

Okay, so first let's do the previous or the in-between question first, which is the different archetypes. I think that's a really interesting question, because if you go on Twitter or you're new, your thoughts are probably that there is just web programming, and maybe there's some other stuff, yeah, like game programming, but you'd be like, game programming in JavaScript and on the web. There's this very myopic view of the programming world, and I bet if you ask a lot of people these days what is the most popular form of programming, they'd probably say web. If you said what contains the most amount of repos, how many percentage of repos on GitHub are web-based, they probably say 90% or some huge number. But the reality is that there's an entire embedded robotics world. You know, you're familiar with the ML side of things. There's networking, there's going to be just performance, operating systems, compilers. There's just huge amounts of variation of all these different types of programming verticals that you can be. And so we often talk about programming in perspective of web, or something that's pretty narrow, and I think that's just a social construct of Twitter more than anything else, that actually I don't believe it's that representative of the entire programming world out there. And I think a lot of programming's really, really fun. There's some really great stuff. Building your own language

is just a very fun experience to do. Every programmer should just do that once, just to have a completely different perspective on how things work in life. But as far as what do I do, I've always looked at myself as a tools engineer. So at my time, at my jobs, typically I would start off on the UI, and then they'd be like, "Okay, well hey, we need a library for this thing." So then I'd be the one writing the library. So in 2012, 2013, I was writing a UI library for the web that can behave just like an iPad, so you can pinch and zoom on it, but it's still a web page. Because we didn't have any of that stuff back then. It was a canvas, had to do all the matrices operations and all that stuff to...

Lex Fridman

Nice.

ThePrimeagen

You know, it felt like you're on an iPad, but it actually wasn't on an iPad. And this was iPad 2 by the way, so this is a long time ago. And so every single time I got into a job it's like, "Okay, hey, we need to do a library. Hey, can you work on a build system?" So back then there was no Grunt, there was no gulp, there was no any of those things, so I had to hand roll my own JavaScript build system. And so I always fell into these positions of building tools for developers to be successful. And I've always really enjoyed that region. So as I went on to say Netflix, spent 10 years there, I'd say the majority of my 10 years were building things for developers to use that they could be successful at their job. And so I've always really enjoyed that aspect, because your shareholders and the people that use your program understand programming and they're going to say, "Hey, I need this." And typically the thing that they need, they actually want. Whereas with people, people want stuff, but what they actually need versus what they actually want often are this weird separation. That's like the old Henry Ford quote, "I just want a faster horse," and he's like, "No, what you actually want is a car." And so you have to play this game of trying to really figure it out, whereas developers, it's like, "I know you know what I'm doing. I know what you want. Let's figure it out together."

Lex Fridman

Actually that gives you a really nice big-picture view of programming in general. So I love the idea of just starting at the interface, like you need to pinch and all that kind of stuff, and then figure out the entire thing that requires to make that happen, including maybe the side quest tooling, how to make it more productive and efficient, all that kind of stuff. So the entirety of the thing. That's really cool.

ThePrimeagen

Yeah.

Lex Fridman

Okay, so that would be full stack? By that general definition of full stack, meaning...

ThePrimeagen

Perhaps, yeah.

Lex Fridman

Versus systems starting at the bottom and trying to optimize a certain kind of specific thing without seeing the big picture of what the resulting interface would look like.

ThePrimeagen

Yeah.

Lex Fridman

And a lot of people in web programming, they never go beyond the front end of how a thing looks. They kind of always assume there'll be somebody, some grunt in the shadows, in the darkness of the basement, that will implement the back end.

ThePrimeagen

Some Gilfoyle out there will be doing the back end.

Lex Fridman

Gilfoyle.

ThePrimeagen

Yeah, I like to call myself a generalist, just to give some ideas. At one point at Netflix I built the WebSocket connection. So for TVs, how WebSocket works is code I just wrote. And so I built the framing thing, and before that I was doing stuff with memory, and before that I built a UI for a tool. Right? I can just do the thing. You just tell me the thing to do and I'll just go do the thing. I don't try to get super good at one specific activity. I don't want to be a Kubernetes engineer, who's the world's greatest employer, but if I had to go learn Kubernetes, I'd go learn it and learn how to deploy some things, and then hopefully move on to the next thing, if that makes sense.

Lex Fridman

I posted about the fact that I'm talking to you on Reddit, and there's a lot of wonderful questions. Somebody mentioned that I should ask you about DevOps. Can you explain what DevOps is? Is it a kind of special ops of programmers, is it SEAL Team Six of developers? What's DevOps? Can you define... Are you a DevOps engineer?

ThePrimeagen

Well, people keep telling me DevOps isn't real. There's actually, you want platform engineers, cloud engineers, infra engineers. I just often think the easiest way, if we're doing just some basic nomenclature, it's just DevOps are the people that make sure that when you launch a service and all of that, it doesn't just disappear. Right? It's all the backbone of being able to

operate something at scale. Really, if you think about it, if you're just writing a mom-and-pa website, people that do PHP that are doing WordPress and all that, they're going to build something, they're going to hand it off to, I don't know, Linode, DigitalOcean, some company. They don't really need a really complicated build, deployment, all this. It's just someone with a simple website so they can sell their goods. And so they don't really need that. And so that's kind of how I think of a DevOps, is when things need to scale, that's the person you hire.

Lex Fridman

Yeah, those people are actually amazing.

ThePrimeagen

Yeah.

Lex Fridman

The time I spent at Google, it's like oh yeah yeah, there's all these fancy machine learning people, but the folks that are running the infrastructure, basically that make sure that shit doesn't go down, they're like wizards, and they're essential.

ThePrimeagen

It's a very incredible vertical of job. And obviously I'm using a very broad term to describe, I'm sure, a bunch... You know, because making sure stuff doesn't go down, you could also say that's an SRE, right? Site reliability engineer. Whatever, the ones that wear the bomber jackets at Google. And so when we say DevOps, I think people get very particular about terms specifically in this category. They're like, "Well actually, you're mentioning infrastructure engineer versus site reliability engineer." It's just like, "Okay, yes, I hear you," but generally when someone thinks DevOps, they think somebody that manages the servers and their life cycles and the reliability. There's DevOps. Is it real? I'm not sure.

Lex Fridman

Okay.

ThePrimeagen

Did Vercel kill DevOps?

Lex Fridman

Question mark?

ThePrimeagen

Question mark.

Lex Fridman

Yeah. Wow, you're almost a journalist. That's a headline. Let's go back to the beginning.

ThePrimeagen

All right.

Lex Fridman

Baby Prime. So you mentioned Netflix.

ThePrimeagen

Oh, I worked at Netflix by the way.

Lex Fridman

For people who don't know who ThePrimeagen is, he mentions the fact that he has been very successful and has worked at Netflix in basically every other sentence.

ThePrimeagen

Correct. Almost as much as I mention Neovim.

Lex Fridman

Oh, great. Tell me more about Neovim. No, please don't. So, baby Prime. At the very beginning. You've had one hell of a life, and I think it's inspiring to a lot of people. You've gone through a lot of painful low points, including meth addiction, loss, and like you mentioned, you've come out of that to become a successful programmer and a person that inspires a huge number of people to get into programming, and just to find success in life. So maybe... I would love it if you laid out just your whole life journey from the beginning.

ThePrimeagen

So I guess if we're going to start with this whole journey, I think it's probably best to start to when I was about four or five years old. That was the first time I was ever exposed to pornography, and it's kind of just earwormed me for a large portion of my life. And so I don't think there was a day that didn't go by from when I was a very young lad all the way up until I was twenty-some years old where I didn't think about porn on the daily basis. And so it was just every single day, even that young. And so it was just a very mind-consuming, time-consuming, thought consuming thing that plagued me, starting at a very young age. When I was seven years old, my dad died. That was a really tough period of life. I still think about this time that I went over to China, and there's some rules that we were given, and one of the rules was just like, "Hey, don't talk about God, and if you do, use the word 'Dad' instead." And I was just like, "Okay, Dad!" It was the first time I said that word in 17 years or some long time. It was so weird to say that phrase. And I was just like, "Oh, that was just the strangest thing I've ever said in my entire lifetime." It just felt so weird.

Lex Fridman

Yeah. Yeah.

ThePrimeagen

So, kind of rewind. As I got older, obviously was very good at computers, good at accessing porn, of course, played video games on the Internet. Fun fun kind of side quest story. I think the guy's name is Lord Toc on Twitch. I can't quite remember his name, but he built this game called Grail, G-R-A-A-L, and Graal Online. And when I was a young lad it was just like Zelda, except for it also had a level editor and it had a C-like language, and that's how I discovered how to program, is I looked at these symbols and figured out what they meant, and then I was able to make things happen in the game. And that's my introduction into programming. So thank you that guy, whatever your Twitch name was. But all right, so keep on going. As I got older, I was super bad socially. I was not a very great social person. High school is brutal, got made fun of a lot, really I wouldn't say had a great time during high school. Definitely felt very out of place or offset or maybe misplaced, if you will. I'm not sure what the right word is. And so of course at that point, I just always wanted to be accepted, to fit in and all that. I did forget to say one side story. After my dad died, my older brother, he started getting into drugs, and along with that he exposed me to pot, so at eight years old I was smoking some marijuana for a while there, until maybe 11 or 12, and took a break, and then again did a lot of that as I got a little bit older, but... So I got a lot of these exposures fairly young. 16, 15 through 18, lot of drinking and all that. When I graduated, or as I was graduating high school, I had such sadness, if you will. I was very sad about how everything went, tried to commit suicide, obviously it was a very poor attempt and I'm still here today. I'm very happy about that aspect. I'm glad that I didn't follow through with anything, had to go to the hospital and all that. And when I was done, I just still remember coming out of the hospital, and at that moment it's kind of like something broke in you. Have you ever read the book Wheel of Time? It's 14,000 pages or something like that, but right around page 12,000, Rand has to intentionally kill a girl, the main character. And that's the moment he breaks, and he gets into like Hard Rand. [inaudible 00:24:46] Rand, if you will, for those that know Wheel of Time will appreciate all that. For those that don't, very confusing, and I understand. Not the Amazon movie show, not that Wheel of Time. So now that we go back onto it, at that point it's just like something kind of broke in me, and I just didn't care anymore. So all the social awkwardness, if you will, all that, just died away with me, but also so did everything else. And so I started using a bunch of drugs. LSD, mushrooms, meth. Did a bunch of meth, did a bunch of that stuff, and then went off to college and continued to do a bunch of stuff. I took too much acid to where for quite a few years, I had little squiggles on the side of my eyes whenever I'd walk by high contrast objects. And so it's just that whole period of life was just kind of marked by just poor decisions. And then sometime when I was about 19 years old, somewhere in that range, I just had this one evening where I felt the very dramatic and real presence of God. And I kind of had this choice, like Frodo, on a razor, where it's like if I go either way, I'm going to fall off, and I need to change my life. You get to make the choice now. Do you want to do that or not? And so I remember going, "Okay, I do

want to change my life. I don't like this experience. I don't like what I'm living. I am still very sad, I still feel very desperate. I still feel all those things. I'm just pretending to be this other person." And then I just went to sleep that night. Nothing changed in my life. Everything was still the way it was. I woke up the next day, the same person, and I was just like, "Oh, that's just such a strange, weird experience." And I just went about my day. And then I remember, I think that evening, I looked at porn, and all of a sudden I just had a conscious... just this deep, profound shame. And I was like, "I've never felt shame in my life. I have no idea what's happening now." And then all of a sudden when I smoked pot, I just felt deep shame. And when I hurt somebody or did something wrong, all of a sudden... It's just like I got a conscious from that evening. That's what my gift was, if you will. And just at that point, I didn't even have a choice. I had to change my life, because for whatever reason, I've been changed in a moment. And so from there I started actually trying in school. I always joke around that I got 2.14 in high school. I had a teacher hand write me a note saying I was the worst student she's ever had. All that kind of stuff. I was not a really great student. And then in that moment it's just like, "Okay, now life's changed," and I start trying to learn, and I try to become a good student. And it turns out it's really hard. I was really bad. I still got Cs. I went and took pre-calculus and failed pre-calculus, and I'm like, "Oh my gosh, I used to be the smart math guy, and now I'm the idiot failing." And so I'm just questioning myself and all that, and I spent hours upon hours in a studying, math learning center, and then just at some point years into this journey, I'm like a year and a half into this journey, at this point, something clicks, and I go from being the worst person to just immediately becoming the best. Everything after that is just, I don't know what happened. All of a sudden I was the best person at math. I started going into my computer science classes. I just really got everything. Everything, at just years after trying, just all of a sudden became easier. And I'm not sure if it happened over the course of weeks or when the easier started, but it was just first predicated by just a huge amount of difficulty. And then this is where I started really desiring and loving the process of learning, was when things started getting easier after all those years. Because I just was motivated by this desire to do something, not thinking it was going to get any easier, and then all of a sudden it just started getting easier, and it was great. And that's really where I guess I started having the biggest parts of my life change at that point. I started really, really, really wanting to never look at porn again, because every single time just such shame, and I really wanted to stop. And that was by far the hardest addiction to quit. Smoking cigarettes was also a really hard addiction to quit, shockingly hard addiction to quit, but porn by far was just the worst of them all. And then I think about 22, I was finally done with all kind of addictions, if you will, and then for a year I just worked in all that, and I think right around, maybe it was 21 and three quarters, somewhere in that range, I'm not really sure where I stopped all the addictions part, but... Or at least the outwardly addictions. And then at some point, six months later, a year later, met my beautiful wife. Things just started falling more and more into place. I loved more and more work. I loved programming. I started programming 12 hours a day. I watched the Social Network movie, and after that, I was just like, "I'm doing a startup." And so that night I started my first startup, and I was just like, so... It was in PHP by way.

Lex Fridman

Nice.

ThePrimeagen

PHP, yeah, 5.2 or something like that. It was great. Great times. And I was just so motivated to do that, and I would just program for... Sometimes I'd program for 24, 36 hours straight, and just nonstop, that's all I wanted to do at all points. I think my wife got a little sick of me. She would be like, "Can you drop me off at school?" And I'd be like, "No, I'm programming." I was not a very nice... You know, I didn't think through things that well.

Lex Fridman

Yeah yeah. Yeah.

ThePrimeagen

I was just so into it and I just did it nonstop, and that's kind of how I became me, is that story, if that makes sense.

Lex Fridman

Let's try to reverse engineer some of the pain and some of the triumph. You made it sound easy at times. Let's try to understand it better, maybe when you were seven years old. What do you think about the pain you've experienced there, losing your dad? What do you think? What kind of impact did it have on you? What kind of memories do you have at that time?

ThePrimeagen

The best way I can put it is that I just never knew what a dad was. I was young enough that I could maybe repress or just even have the capability of remembering things long-term. Because I know most people don't remember a lot from when they're young, and so I'm not exactly sure. I probably was at one of the best possible ages, if I'm going to lose a dad, to lose a dad. You know? If you're going to lose one, if you're 11 or 12, it's a terrible age. That's what my brother was, and he fell into drug addiction and never got back out. So I just have more of a fuzziness and just kind of a longing. I just wish I had a dad.

Lex Fridman

What impact did that have on your evolution, on your life, having that longing?

ThePrimeagen

I think that's why I was so bad socially, in the sense that I was looking for approval, right? I needed approval. I think a lot of people desire that approval or that loving figure, and I just didn't have that. So I think I just looked for it in everything else, right? If I were to psychoanalyze my actions. During the time, it's not like I was actively thinking that, but yeah, I just always wanted something to fill in whatever that was I felt.

Lex Fridman

I think a lot of people listening to this will resonate with your experience in high school. Being the outsider, being picked on, struggling through a lot of different complexities at home. What advice would you give to them?

ThePrimeagen

The worst part about high school is that you're surrounded by a bunch of people your age and it feels eternal.

Lex Fridman

Yeah.

ThePrimeagen

You don't think... The people that are around you, you feel like are the people that will be there for the rest of your life. At least that's what I thought. And I didn't really even realize this until many years later, that they are going to be some of the least consequential people in your life.

Lex Fridman

Yeah.

ThePrimeagen

Which is very shocking to think about, especially if you're in it right now.

Lex Fridman

Yeah.

ThePrimeagen

Right? Right now they are everything that your experience is, your whole reality. And then one day it all stops, and then real life starts to begin.

Lex Fridman

Yeah.

ThePrimeagen

That's such a shocking thing, and if I could just tell myself that, maybe I would have been a bunch of different person.

Lex Fridman

That's so beautifully put. I mean, it is like a trial run. You know at the beginning of video games, there's a little tutorial? That's what that is.

ThePrimeagen

Yeah.

Lex Fridman

And actually that should be a chance to try shit out, to take risks, because real life will begin with, there is more consequences after that.

ThePrimeagen

Yeah.

Lex Fridman

Because real life will begin where there is more consequences after that.

ThePrimeagen

Yeah.

Lex Fridman

Here you can, if you like a girl, ask her out. Try, try shit. If you get picked on, hit that guy back. Try shit out.

ThePrimeagen

I'm not going to condone punching another person.

Lex Fridman

I will. Beat the shit out of him, and take some jiu-jitsu and learn how to take him down. And then that girl that rejected you will be like, "Hmm, maybe I'll give that guy a second chance." Be a bad motherfucker. It's a chance to try stuff out. This is a very motivational speech for kicking ass.

ThePrimeagen

It is true. I mean, there is something very true about that, that I think especially... I mean, I have no idea what the girls experience of high school would be like, but as a guy, there's definitely a lot of like physical requirements in high school. There's a lot of physical measurement, at least where I grew up. I think that might not be true in all high schools, but if they're filled with boys, it's probably true. And so it's just like, yeah, it probably does help to do those things, to go to BJJ, to do any of these activities. Because even if you don't ever kick someone's ass, just having some level of confidence in yourself is probably a very valuable thing. But just remembering that this is such a short, tiny moment in your life is just like a huge help.

Lex Fridman

I mean, the way you phrased it is exactly right. That's what it feels like. That these are the people that will be with you for the rest of your life and this is the whole world. And so that means that there'll be just tremendous amount of impact if somebody picks on you or if you fall somewhere low in the hierarchy and the status hierarchy of this high school, that means you'll be low in the status hierarchy of the world and you're fucked for the rest of your life. And that carries a tremendous amount of weight. It's just why psychologically it's extremely difficult to be... I think it's understated often by parents, by society, how difficult it is to be a high schooler, how difficult psychologically it is, how it actually makes sense that some people would suffer from depression and be on the verge of suicide; is very, very difficult.

ThePrimeagen

Yeah, I think it's even... People always say, "Back in my day," blah blah blah. I think it's genuinely harder today than it's ever been in the sense that when I was a kid, there was a qualification to people. Meaning, this is a cool guy, this is not a cool guy. Today, there's a quantification of people. You have 32,514 people following you, you have 12. The people can visually... They can inspect your exact social value on whatever platform you're on. And that has to be just so much harder. And I can imagine that there's a lot of just so much weight to put on that, that it's just... it feels probably way worse and way more damning to be uncool because you have an exact number of how uncool you are.

Lex Fridman

Yeah. The challenge there. And the task, the quest is to remember that just because your social circle on social media and in high school thinks you're uncool, it actually might mean you are cool. And you need to find that cool and grow it and let it flourish so that when real life begins, you can fucking come out of the gate firing on all cylinders because-

ThePrimeagen

That's a great way to put it.

Lex Fridman

I think if anything, high school is really bad at picking out the cool people. Whatever the system, the hierarchy that forms, it's such a basic bitch hierarchy. You're good at very generic shit. That's how you rise.

ThePrimeagen

Your parents bought you an expensive car.

Lex Fridman

Expensive car, right?

ThePrimeagen

Just -

Lex Fridman

Materialistic shit. Yeah, exactly.

ThePrimeagen

It's a greedy search. See, they didn't have a proper search, so they're just hitting that local optima.

Lex Fridman

But the... I mean, even the objective function for that greedy search is just a really shitty one, where those people that win the game of high school are very often not going to be the people that win the much more exciting, beautiful game of life. So do epic shit and try stuff out. The weirdos are the ones that are going to succeed, the weirdos in high school. Probably because they also get bullied and they get to be tormented more psychologically and get to explore their own mind and think through what it means to be a human being more. Because if you're winning in high school, you're not being challenged, you're not self-reflecting, you're not trying shit out. So there is some degree to being tormented as long as it doesn't break you. The porn addiction, that's another powerful one that I think will probably resonate with a lot of people. And it's interesting that you say that's one of the hardest addictions to overcome.

ThePrimeagen

Let me say it this way, some addictions have a much bigger societal look and porn is just not one of them, which makes it super hard. None of your friends are going to cheer you on. If you go on Twitter and say, "I quit porn," they're going to be like, "Well, that's good for you but not everybody..." No one makes that argument with meth, right? No one's going to be like, "Well, not everyone has to quit meth, okay. It's actually a fine industry and people who are the ones producing it, they're good also, right?" No one's going to make that kind of argument. Whereas with porn, you're going to have a whole thing and friends are going to think you're dumb for doing it or whatever. It's like you have... It's a much more difficult one in just like that. So it feels accepted.

Lex Fridman

And I think it's also an addiction you can practice, participate in privately and hide it from the world. There's certain addictions that are harder to hide from the world for prolonged periods of time. And porn addiction is probably one you can just have for many years and then it can deepen. That's probably a serious issue. Boy, am I glad I grew up before the internet because porn is so accessible, so easy to go deep into that addiction. I mean, what can you speak about what impact it had on your life? Maybe some of the low points, but also how to overcome it?

ThePrimeagen

I'd say as far as impact goes is that you will have such a long and broken look at women. By the very, like I can... Again, I'm only speaking from a male's perspective, that porn in its just most basic thing is that you use another person for your own desire or your own want. It's not something that is deeply needed. There's no need for porn. It's purely a want-based activity or a lust, however you want, whatever word you can fill in there. And it is purely an objectifying activity. Someone else is on display for your own enjoyment. And so I think you carry this around. I do think that the women that I dated during high school or the women after high school and college, I looked at them as a means to an end. And I think porn greatly kind of shifted that perspective in my head that I did not give the value that was desired to another person. It really devalues humanity just in general, is my perspective of it. And then it makes people into commodities. And I don't think people are commodities. I think everyone has value. And so during that, for me that's like the great effect of porn, is that it's just consumerism gone wild or materialism maybe, you could ask or argue, gone wild. And it's extremely hard to quit, just like you said, because I can look at porn and then I can go out to lunch. No one's going to know. No one's going to have any ideas. It's a very private, it can be very short session. It doesn't have to be something that takes... You can't take acid then go out to lunch, right? Your whole day is going to be a very different day. And so it's very quick, easy, accessible. And then obviously there's all the science and statistics, like men make worse decisions for some period of time after looking or being exposed to sexualized images. There's the whole dopamine effect that's just like you constantly need more and more dopamine. That's why people typically don't just watch five minutes of porn and call it a day. There's like the hundred tab joke that's always made on the internet. It's because it's just this constant dopamine cycle you're constantly doing. And all that stuff is great to say. And I'm sure statistics and science and all that stuff is really great arguments for some amount of people. But for me it just comes down to, is it really a good thing to do? Is it really actually something we want, is to value people in such a profane or just disregarding way? I just really think it's just bad for the soul. Even if all the stats said it was great for you, I still say it's actually bad.

Lex Fridman

Yeah. You have to look at the long-term big picture, psychological impact it has on your relationships with human beings in general. That's my, more generally than just porn, my problem with the quote, unquote, "sort of manosphere", is I think sleeping with a bunch of women is great, wonderful. But the problem is making that the primary objective of your life, similar with porn, is you devalue one of the most awesome things, which is intimacy. That's true for deep friendship, that's true for relationships. And I think porn does that in its purest, darkest form, which is: the thing that matters is the sex, not the deep connection with another human being. And I think, again, going back to high school and the manosphere, the objective function if it's to get laid, which helps with status and confidence and all... All that is wonderful, I think. Again, can be an addiction. But the thing

that's even more awesome for a lot of people is a deep friendship or deep intimacy with a romantic partner. That's also fucking awesome, and both of those are great.

ThePrimeagen

It's objectively better to have... I would say that there's no universe that exists or there should be no argument possible that exists that a guy who has meaningless sex has a better or a more meaningful life than, say, me and my wife who've been together for 15 years. We have a very... I can depend on her in all circumstances. Whereas if you live that other life, it sure could be... It could feel great, but there's no meaning to it. There's no actual real value to it.

Lex Fridman

That's absolutely correct. I do think that getting laid can have a tremendous positive impact on the confidence of a young man. I think just there's a certain number of sexual partners from which you can collect a lot of data and it can free you about, like not to be so nervous about the opposite sex, not to be so nervous about human interaction. And that will allow you to see the world more clearly and to actually find that one partner with whom you can be deeply intimate with. Sometimes the nervousness around this societally constructed value in getting laid can cloud your judgment. And if you just release that by getting laid a bunch of times, then you could see the world clearly that getting laid is not nearly as important as you said, as finding the right human, including I should put in that pile, not just a romantic partner, but friendships, deep lasting friendships.

ThePrimeagen

Well, I mean I think you're right that our society puts a lot of emphasis on getting laid. And I'm sure that's true among any group of males throughout any point in history. I'm sure that's a very common joke that's never actually never stopped at any point. So I'm sure that exists but... And there's probably some truth to the sense that after you've... Who was it? Jim Carrey. "I hope that everyone can get rich so they realize that money solves none of your problems." The realization that this thing that society told you is hyper important is actually not the important part. It is a very important... It's a great sign that your relationship is healthy. Like if me and my wife were to have no sex at all for months on end, something's gone wrong, which means what... we are no longer on the same plane. But it's not also a good identifier. Just because you're having a lot of sex, it doesn't mean you're having a good relationship. And so it's like a unique... I forget the right term here, but it's a unique way at looking at the problems. And our society puts so much emphasis. And maybe that's why porn was so hard to quit, but my guess is it's just all the dopamine effect that it is. But for me, the most important part and the thing that actually has real reward is having that... having just my wife. I do not look at... I desperately try not to look at any other woman. I'm hopefully not going to get caught... Mark Zuckerberged at the White House like that. I don't look at porn. My wife has complete confidence in me that there is not going to be a situation in which she has to question me in any kind of sense. And that builds a much more deeply, I

would argue, a very deep relationship because the trust is that much bigger. I think the deepness of the relationship is probably proportional to the trust you have in each other. It's very hard to have a deep relationship with no trust.

Lex Fridman

Yeah. And a probably a prerequisite, maybe a component of trust is vulnerability to where you take the leap of being vulnerable with another human being. And that vulnerability when reciprocated builds this really strong trust and it's a beautiful thing. Yeah. I personally just... Given my position, that's even more challenging, being vulnerable with the world and there's a bunch of people out there that want to hurt you for it, but I think it's worthwhile anyway to be vulnerable.

ThePrimeagen

It's always worth it. The risk is always worth it in some sense. Obviously, everyone has a different life they have to filter through their actions with, right? Because the person that has no, say, social following or anything, their risk reward profile could just be local impact, which could be just as damning or harming to them. And so it's always worth the risk though, in my personal opinion, because finding my wife has been obviously the most impactful or changing thing in my life. Or second most, I'd argue that one night with God would probably be the most impactful thing that led to everything else, but then the wife would be the next most impactful. I mean, I'm cleaning up after myself and stuff now. Changed man. I'm a changed man.

Lex Fridman

Can we try to reverse engineer that moment of you finding God. What is it at 19? Because it feels like that was a big leap for you to escape the pain, to escape the addiction or the beginning of that journey. What do you think happened there?

ThePrimeagen

I think I just felt like I just... There was no line that I wasn't willing to cross. Everything was fine and just like... It just all of a sudden, just in that moment, it's just like I had I guess some sort of deep fear and understanding I am going down a path. Is this really the path you want to go down? And I don't know what the result of that path would be or anything like that. I don't tend to speculate on things I don't understand. I just know that in that moment I had the option and I just chose... I didn't want it anymore. Right? It's kind of mixed in this whole thing where it's just like I had no value. I wrapped up all my meaning or value in having sex or getting laid, I had... All that stuff, all the things we just talked about, that was where all my worth was. And that is just such a terrible place to have your worth. And it was just all came to a point. And I can't tell you the day of the week, I can't tell you anything other than it was nighttime and I was in South Hedges in Montana State University, go Bobcats, that's about... Yeah, that's the sign that we do at football games. Don't worry about it. But that's all I can

really tell you because that night was no more or less special than some other night. It's just the specialness was I got at least a chance to make a choice.

Lex Fridman

Because you find in that advice that you can give to others who are probably... There's probably just an endless amount of people that are struggling with porn addiction now, young people. What advice could you give to them? How to overcome it?

ThePrimeagen

For me to overcome it, I had to realize that I was taking something away from my future wife. Some people would be like, "Oh, well, once you get a girlfriend then you can stop." And it's just like, "No, because you never stopped the problem." You don't stop a problem by replacing it. And so I didn't have a girlfriend, I didn't have all that. I just realized that I was truly taking away from something from my future wife. And I didn't even know my current wife at that time. She was not in the picture. I'm not even sure if she was at Montana State University at that point. And so it's just that's... Once I made that realization, I think it went from my head to my heart, which they say is the greatest distance in the universe. I finally got it. And that's really where things change. The ability to say like what's going to help you change and all that, I don't know if there's... I don't think there's silver bullets, right? If someone could offer you a drug... I forget who says this phrase, but there's this really interesting phrase that goes something like, he was a very depressed man and he was struggling with suicide and he writes about this in this memoir. And he goes to these doctors and the doctors effectively say, "Well, here's antidepressants, it's going to help you." And he says that, "Well, the problem was is that scientists told me that I could just touch my brain and make myself happy, and that's it. They could reach in, they could configure some stuff and I'll be happy." He's like, "For me, it was a lot like going out into a field and being able to take a drug to see the rain. I could look out, see the rain, it would fall down, it'd be silvery, it'd be beautiful, but all the crop would still die because there's not actually any rain. I had to discover how to be happy myself." And so for me, it's like the reason why I looked at porn is because I was unhappy. I was trying to find meaning. I was trying to find value in something, right? Something that was supposed to finally give me this ultimate satisfaction. And it just does not, no matter how hard, and no matter how much you think it will, there is no escapade, there is no pornography that will ever give you that satisfaction you're looking for. That's the reason why it's addicting. And that's my call to why you shouldn't do it, but how to get out of it, I only got out of it by realizing.

Lex Fridman

I think that's really brilliantly described. You knew that this thing you're doing is preventing you from finding your future wife and future wife could mean more even broadly, this path to a flourishing, to a beautiful life. I think there's a lot of choices we make that are just preventing us from opening the door to whatever future. I think what's really nice to do is to imagine, just like we said with high school, that there are a bunch of trajectories in life

where you'll be truly happy and you need to construct your life in a way where you have the chance to travel down those paths. And there's a bunch of addictions, there's a bunch of choices that prevent us from traveling down those paths. So just believe that you're going to have an awesome life and remove from your life the things that are preventing you from walking down that path, which is essentially what you did. It's a leap of faith that if you let go of porn, that a better life is waiting for you on the other end.

ThePrimeagen

Yeah. I definitely can't say how long it will take, a better life. But for me, there's no way in the universe I could have had the relationship that I have without first making those steps because I couldn't value my wife in the way that was proper for who she was. I would have valued her through the index or the lens that I currently was looking through.

Lex Fridman

Got to ask. So I've never done meth. I've never done meth.

ThePrimeagen

That was a great segue by the way.

Lex Fridman

Oh, man. I don't know what the fuck I'm doing, honestly with this interviewing thing. But yeah, meth and LSD... I did ayahuasca. I did shrooms a bunch of times. And this topic, I should say that there's a lot of, on Twitter and in the tech community in general, people speaking negatively about ayahuasca and some positively. I think it's such a roll of the dice. I had incredible experiences, but I don't think I want to recommend it to anyone. It's a risk, it's a serious risk. It really is a roll of the dice that you could meet your demons and they could destroy you or you can meet your demons and let go of them. Or you could have experiences like I did, which is never... Apparently I don't have demons. I'm pretty sure they're somewhere in the basement, but I've never met them on drugs. I'm always really happy. I'm happy drunk. I'm super happy on ayahuasca, just full of love. I don't understand, I don't understand where the demons are, but that's my biochemistry, whatever that is. And for some others, one trip could be amazing and the next one could just completely destroy you and wreck your life. So I don't know what the recommendation from that is, maybe avoid it, but then all of us die and life... I tend to lean into adventure but drugs is... If you fuck with the biochemistry of your brain, you can really destroy yourself in a way that it's going to torment you. So I would generally recommend that people avoid drugs altogether, probably, unless you're a crazy motherfucker. Hunter S. Thompson.

ThePrimeagen

What an intro to this topic.

Lex Fridman

I'm sorry. What's meth like?

ThePrimeagen

That's a great intro. I like... You are very correct in the sense that there is, at least when it comes to hallucinogens, there is a wild variance to what you're going to experience. And there is no guarantee, there's no... Just because you buy the product, it doesn't mean you're going to have a good time, right? There's a lot of... Personally, I find that stuff to be very... I believe in the spiritual realm, right? I believe demons and angels exist. I believe God exists. And that whole realm is like... I don't know what it opens you up to, but it's much, much different experience. Now, some people will be like, "Oh, it's just a bunch of chemicals in your brain. They all get mixed up. LSD just takes all of your pathways and they all go... They all get kind of scrambled up in your brain." And it's just like, "Yeah, the experiences are profound." I had some really bizarre, very cool, very awful... I've had all the experiences in them all. I can just tell you that I personally always say the same thing. It's like, choices that I made I can never take back. I would never take that away from myself because I don't know if I would be who I am today without all those experiences going up to it. But if you have not had that experience, I'm on your team, or at least partially on your team, maybe more severely, I don't think you need those experiences. I don't think they're going to... You don't have to put yourself through that to make a good decisions or to realize that people have value, right? You don't have to do that. So as far as what is meth like? Meth is like... If you've ever done cocaine, cocaine starts off with like a 15-minute dance party. Just... It's just so intense. It's so great. And then it just followed up with like a five hour... just feeling wiggly, right? I don't know how else to describe it. Meth is like that except for I didn't get as much dance party or any dance party, but instead I just got that part for like 12 hours.

Lex Fridman

Yeah.

ThePrimeagen

So did a lot of skateboarding, did a lot of running around, staying all night.

Lex Fridman

Would you say it's a pleasant feeling or is it more like an escape from the loneliness of life? Is it pleasant or negative in the actual moment? Not the consequences but in the moment.

ThePrimeagen

So I mean, this is just a very interesting kind of area, which is that not... Universally, you can't say that. Often you'll find that there's kind of these two groups of drug addicts. There's those that like the opioids and those that like the uppers. They typically don't like... There's very few people in the drug world that do both. They're really just like find their side and they go for it. So is meth a thing that everybody's going to enjoy? Well, categorically, as you can

see, and just how people experience drug addiction, no. But for me it's just I had a really... It kind of feeds into the ADHD nature of this... Because you know you're kind of high energy, you're like always in the moment. So it's just like you're in the moment, but it's just like, "Oh, I'm in the moment!" Everything's just so intense! You just want to really be in the moment. And so it's just experiencing that constantly. And so was that great? Well, some people... My wife always tells me this, being nervous or... I forget, the anxiety of a situation can also be the same thing as like thrill. I forget the exact way. She's probably super disappointed that I messed this up. But it's like you could perceive those two experiences in very different lights. Some people get in front of a crowd and it's thrilling. Some people get in front of it and it's just the worst experience of their lifetime. They would actually literally rather die, which is a crazy thing to think about than stand up and speak. And so for me, meth was that thrilling side, but at the same time, it still didn't quite give me that thing I wanted, whatever I was looking for. I'd use it to help try to get that thing I want, but it was never giving me that thing I wanted.

Lex Fridman

Yeah. For me, I've had all really wonderful experiences. Do not recommend them. But like with shroom-

ThePrimeagen

That's a YouTube policy by the way that you have to say, "By the way, do whatever you do, do not do a illegal activity."

Lex Fridman

I-

ThePrimeagen

But I had great experiences, but whatever you do, don't do it.

Lex Fridman

Mr. ThePrimeagen, I have no master. I don't have YouTube or whatever. I'll say whatever the fuck I want. I'm just-

ThePrimeagen

But seriously, you do.

Lex Fridman

... kind... No. No, I don't give a shit about YouTube or anybody, honestly. I'm just careful about the words I say because just because I had positive experiences, I don't want young people listening to this think they should try the experience. I think the much more powerful message is that life is awesome even without that. That's something I definitely experiment with on the alcohol side. So for me, I'm an introvert. I'm afraid of the world. Social

interaction fills me with anxiety. Alcohol is definitely a thing that helps with that sometimes, but I think honestly it's not even the alcohol, it's having to do something while a person is talking to me. I could just drink a liquid. "Yeah. Mm-hmm." There's a social thing. With a beer, it's like... "Yeah. Uh-huh. Yeah, we're having fun." And I think it's... For me, it works the same as... If the liquid actually looks like alcohol, it does the same purpose often because alcohol... If you have a whiskey or a beer looking thing, it kind of sends a signal that we should be having fun. So we're socializing, right? We're fucking getting crazy. And then that mean... You don't actually need the alcohol. You can get fucking crazy without the alcohol substance, but there is some kind of social signaling that happens when you have a drink in your hand. So I've been to get-togethers where I'm not drinking, but just doing a fake drink situation and I can also have fun. So I've been... But that said, traveling across the world, there are times when you get to be able to down a bottle of vodka. That's very essential for my line of work, but that's almost like a cultural experience versus a necessary component of a successful social interaction and one that brings you happiness. So not drinking... I think you can have fun and not drink too. So all of this... Man, I'm so careful saying drugs have had a good effect on my life because I think for most people, no, for majority of people, they will in the long term have a negative effect. So I think if you were to choose one or the other, just no drugs and no drinking means one day you can be the President of the United States kids. And I should say... Oh, man.

ThePrimeagen

That is-

Lex Fridman

That means Diet Coke-

ThePrimeagen

... his funniest line.

Lex Fridman

Diet Coke is great.

ThePrimeagen

That's his funniest line, which is, "You would hate me if I drank." Which I just like... To me, that tickles me to no end. Just like, "Oh my gosh, that is such a funny line."

Lex Fridman

Self-awareness and humor is wonderful there, but yeah.

ThePrimeagen

But I am on your team. All of the reasons why I used drugs and all that, it's some level of escapism. I'm sure that's like... would be the archetype or the box I'd put that into or the

pursuit of trying to feel something that cannot come from them. It's like trying to find meaning in your job. You can find satisfaction in what you do. That is a very good thing. You can find satisfaction and be happy with what you've created. You can be thrilled by the experience, but you cannot find... I doubt you can find purpose. Maybe some people in specific jobs. This obviously have very broad strokes, I'm painting with. Like if you're an EMT and you save someone's life, maybe there can be purpose in that whole experience, right? So I'm not saying all things, but as programming goes, most programmers, you cannot just simply find your purpose. And same with drugs, you cannot find that thing you're looking for, but they are a very great distraction. And then at some point that distraction comes with a heavy cost. I think Dr. Faust would probably know the best about the heavy cost, but it's just you're making one trade for another and at some point the bill comes due and that bill can be very, very large.

Lex Fridman

The other moment you mentioned that I think is really inspiring is that you failed pre-calculus. You really struggled in school. You realize that school is really hard and then eventually you're able to sort of persevere and, I don't know, break through that wall of struggle. Can you, by way of advice, figure out what happened and what kind of advice you can give to people who are struggling?

ThePrimeagen

Yeah. I'll paint it in a more clear picture, or a very fast speed run of it is that I took pre-calculus, failed. I took pre-calculus again, failed, took pre-calculus again and got a C. So I took it three times. Then I took Calc over the summer, so Calc 1. In that one at the end, the final... The final was a two-hour final. I finished it in 30 minutes and that was the highest score in all of the school. And I proceeded to be the highest score in all calculus and Diffy Q. I was the only person out of 400 people to finish the Diffy Q final. And I got the highest grade. And so I was like... I got really good. So I somehow went from really bad to really good. And my only... The thing that I did is that I had to win. It was not a option. It was not like, "Oh, this would be really great." It's like, "I will not graduate, I will not finish my stuff if I cannot do this." And so every single day I got up, I went to my however many hour class it was. Right after that, I went straight to the math learning center, did those problems. When I got home, I just got the book and it had the odd answers in the back. And I would try to walk through the problems over and over and over and over again until I absolutely got it. And it just became this thing where it's just I... Just simple rote memory took over and the ability to just effectively have the times table, but for calculus, all stuck in my head. Inverse trig substitution, trig substitution, doing Taylor and MaClaurin series. All those things, just over and over and over and over again. Eventually they became easy. They became very easy. It's just that I had to cram it in there. And some people, you hear these stories, whether they barely show up to class and they get As, I've never been that person. I've always been the person that has to sit down, read through everything, and I'm bad at abstract concepts. I like the concrete into the abstract, not the abstract into the concrete. Very bad at talking

about things theoretically, then trying to apply them. But if I can do it once literally, then it's really easy for me to go into the abstract. And so it's just like... For me, it's just I had... There's no substitute for the hours. So if I were to give advice, it's just that you have to have time in the saddle. Hour after hour will make you slowly better. And at first, it's crushing. It's defeating and it's not fun because you are bad at it. But then at some point you're just not bad at it if you can just do it long enough, and you'll start getting okay at it. And then at some point you might even get good at it. And when you get good at something, it feels amazing. There's like an exploratory thing. If you've ever played a musical instrument, you stop having to think about all the little teeny things you have to do to be able to play something correctly. And you start thinking about how you can explore that space. ...play something correctly and you start thinking about how you can explore that space. It's like it's a completely different problem. And same with programming, programming has an identical kind of feel to it. It's just like you'll cross that barrier and it becomes magical as opposed to a chore.

Lex Fridman

Yeah. Once you cross that barrier, somehow other things become easier. But then if you want to have a truly successful life, then you find the next barrier. Yeah, I've always been the same. Everything's come really hard.

ThePrimeagen

Yeah, I've had no free lunches. Everything's just been a lot of pain and struggle.

Lex Fridman

I think somebody said that on this topic that you think work smarter not harder is a phrase that you dislike. Somebody on Reddit told me this.

ThePrimeagen

Yeah. I don't just dislike it. I hate that phrase.

Lex Fridman

Okay. Tell me about your hatred. How do you feel?

ThePrimeagen

The reason why I dislike that is that there is kind of a hidden suggestion there, which is that you already know what smarter is, so just do that. That actually things should be easy. You should just not have to try that hard. You should just do the quick, easy, obvious path and boom, it's done. It's like I've never experienced that in anything I've done. Everything is actually really hard and most of the time I don't even know what I'm doing, so therefore I don't even know what smart looks like. And so for me, the only way I can learn how to work smart is by working very, very hard and knowing that there's no shortcuts. And then when I finally figure out what smart is, when I work smart and work hard, it is that much better.

Lex Fridman

I think there's a deep profound truth to that.

ThePrimeagen

There's a lot of these phrases that just drive me nuts in our society,

Lex Fridman

But that one is... Sorry, that one is really accepted if you can just linger on it because it really bothers me as well. So one, which is a really nice thing you said, the presumption there is things should be easy and you're a failure if you don't see the easy path. That's kind of the implied thing.

ThePrimeagen

Just work smart, daug, why are you putting in all those hours?

Lex Fridman

And so it makes a lot of people that struggle feel like they're a failure because I don't see it. And then the choice they have, well, I'll just be lazy and then maybe the profound truth will come to me somehow. And yeah, I don't think I've ever, and I don't think I've met great engineers that find the smart way without the extremely hard work. The annoying thing about those great engineers is then looking back, they forget the hard work because they remember all the joy they now are experiencing from all the efficient, smart work they figured out how to do. They forget. So when they give advice they give the stupid advice of, well, just do it like the easy way And here's the easy way. But no, you have to put in the hours. Musical instrument is a beautiful example of guitar and piano. I've put in, I don't know how many thousands of hours. And now when I'm explaining stuff jiu-jitsu as well, I sound like one of those people just relax in jiu-jitsu. By the way, just relax is a really wonderful thing for physical endeavors like piano and so on. But to learn how to relax your hand, how to relax your mind, your body and use whatever the biomechanics of your body to apply the correct kind of leverage and the timing and all that, that takes thousands of hours of learning. Just to learn how to relax takes a lot of really hard work. In jiu-jitsu that takes many months of getting your ass beat over and over until you ride the bus home crying. Your ego completely shattered and destroyed. And then a little element is figured out late that night or next morning. And from the depression, there's this little plant that grows this flower of insight. And you use that insight to then get your ass kicked again all next month and year. And then you grow and grow and grow. And from that you discover how beautifully simple jiu-jitsu is or Judo is, just speaking for myself, or piano or guitar. And then yes, the profound truth or the mastery of a skill feels simple when you finally arrive to it, but the path for most people is going to be a hard one.

ThePrimeagen

I think I should make an addendum to the phrase, I think the phrase should be work hard, get smart.

Lex Fridman

Nice. That's a t-shirt.

ThePrimeagen

That's what it should be.

Lex Fridman

Yeah, agreed. Okay, that was a tangent of a tangent.

ThePrimeagen

Can I say one more cultural phrase that I absolutely hate?

Lex Fridman

Yes.

ThePrimeagen

The journey is better than the destination. Everyone's heard this. Just take one second to apply what that means. That means forever starting from now, you are only going towards a place that's worse. That literally is what it means. Enjoy the journey, celebrate the destination, that should be what it would be but no. People say these phrases, they're everywhere. There's these very shallow phrases that have no logical bounds to them. You're just like, why would the journey ever be better than the destination? I think this might even be a C.S. Lewis quote is that C.S. Lewis was like, nope, this is terrible. The journey is not in fact better than the destination.

Lex Fridman

I love the demotivational posters. Progress, moving forward is better than moving backwards even if you're still going nowhere. There's a lot-

ThePrimeagen

I feel that one so much being in California for a few years, that is painful.

Lex Fridman

Positivity, if it doesn't break you today don't worry, it will try again tomorrow. It's just a lot of really great posters.

ThePrimeagen

I didn't even know this was a thing.

Lex Fridman

This is a thing.

ThePrimeagen

Oh my gosh, I want that.

Lex Fridman

Yeah.

ThePrimeagen

Hey. Hi, this is ThePrimeagen. One thing that I forgot to mention in this podcast, which feels just so foolish to me for forgetting, is just what a big role my mom played in my life. She had to work 18 hours a day after my dad died. She really made her house be able to survive. I always looked up to her and I always thought her amazing. And she really was the reason why when I decided to get my butt kicked back in gear, she's just someone who I looked to as an internal inspiration for me to continuing, to keep on going because I really wanted to make her proud. And all those years of just high energy effort, I really wanted to make sure that she knew that I was just so dang appreciative for it. So hey, I just wanted to say thank you. Love you, mom.

Lex Fridman

For people who don't know, you worked in Netflix.

ThePrimeagen

By the way.

Lex Fridman

By the way. Now, how did you go from there, from the hardship that we mentioned, from the struggle, from the addictions and so on to a place where you were working at this incredible engineering company and building cool shit there? So tell the Netflix story.

ThePrimeagen

Yeah, so I kind of alluded to it earlier that I wanted to do my own startup so for, I forget how long it was, one or two years or two and a half years, built a startup. PHP, jQuery, everyone's favorite language is all put together. You can solve math stuff with jQuery. So I just was totally into just non-stop doing that. This is the height of Stack Overflow. I was asking really dumb questions on Stack Overflow like what is more pythonic? And then you get a bunch of up votes and try to steal a bunch of karma away, all the fun stuff to do. Good times. And I was just so into it breathing and I just breathe it in, breathe it out, and that's what I do all day every day. And so it's just like non-stop building of a startup. Ultimately that startup failed and so I had to go get a real job.

Lex Fridman

Can you say what the startup was?

ThePrimeagen

It is so wild thinking about it in the past. Before I tell you what it is, I want to tell one quick thing about my dad. My dad in the early '90s, like '91, '92, was building kind of a phone card company where you'd be able to pre-purchase long-distance minutes. Now, if you remember the '90s at about what '97, '98, '99, 10-10-220, all those different things down the center, all those companies where you can pre-purchase long-distance minutes kind of came out and were very, very big. And so my dad was six years early to that notion and ultimately his startup failed. But he was just really early to something that would catch on really, really big specifically in the telecommunication space. Me as I grew up and did my own startup, I did a startup where was text message marketing. This was in 2010 where you could receive, say texts about various deals, all that kind of stuff. And of course, 10 years later now you don't stop receiving texts and text message marketing is all the rage. And so I also, much like my father had a startup in the telemarketing space in which was just like a half decade too early.

Lex Fridman

So is it fair to say you're almost always ahead of your time, that you're a visionary of sorts?

ThePrimeagen

No, in fact, I am not ahead of my time. I just got, some would say I got unlucky on that situation. But it seems so obvious to me at that time when I was doing it, 80% of phones were dumb phones. Most people had flip phones. When I went and sold Via Text, is what the name was of that specific product. And we had the short code via text too, so it was pretty clever, six digits. When I went out and sold it, I only had a flip phone during that time. I didn't even have a smartphone because they were kind of untenable for a lot of people. So it's kind of just wild times to think about. But then after that, obviously had to get a real job. We were living in an apartment right next to campus, Bozeman, Montana. And the guy below us must've been on some amount of drugs. He threatened to kill us several times, would just scream and just lose his marbles all the time. Very unhinged man, angry downstairs man is what we call him. One time my wife dropped a battery, double A. Okay, so we're not talking about a B battery or D battery. We're just talking about a double A, drop it pa, land on the ground, "I'm going to kill you." Like crazy, absolutely unhinged behavior down there. So I had to go get a real job, we needed to move out of there. We were going to start our life. And so I worked at a small place [inaudible 01:16:38], which I kind of talked about the boredom there. Got to go to a place called WebFilings where I'm working just tons and tons of hours. During all that time I'm still trying to figure out startups. Did one where you could pre-wish your friend's birthday messages, and then it would automatically send it via Facebook beforehand. We called it Greet Feed. It was pretty clever. Nonetheless, I say all that story because everything that I was doing was exploring, building, finishing things, working,

learning about corporate life, learning how to communicate in corporate life, being able to be successful at a job, learning about a bunch of technologies that we're about. And one of the big technologies during that day, specifically 2013 was RXJS, if you remember that one, RxJS, that's a link from C# kind of ported over to JavaScript.

Lex Fridman

And for people who don't know, I guess C-sharp, what is its closest neighbor? Java. Is Java-

ThePrimeagen

They obviously just took Java and ripped it off at one point, but now it's such a dynamic, interesting language that it seems like it could be a really cool bounds of practical versus not practical. It's just I'm not really into wearing pleated pants and programming at a Microsoft house.

Lex Fridman

Is pleated pants a requirement?

ThePrimeagen

I think so.

Lex Fridman

Okay, we'll get back to this.

ThePrimeagen

Can we just get back-

Lex Fridman

All right.

ThePrimeagen

Triggering me here.

Lex Fridman

WebFilings.

ThePrimeagen

So anyways, WebFilings was that's where I had to do all the matrices stuff and build systems and just kind of all that. And it really pushed me because they also wanted me to do 60 hours a week. It was not very healthy work-life balance. It was very hard work. And kind of that really hard work going to cutting edge stuff, really understanding the world, really made it so that I was able to just be able to talk about stuff very commandingly because we had to build really complex state machines for the UI for what we're building. And so when I went

and started getting a LinkedIn and all that, inevitably just due to the fact that I've touched all these technologies and I had some sort of paper trail saying, I've touched these technologies or Microsoft. Dang it, Lex

Lex Fridman

Pleated pants.

ThePrimeagen

Pleated pants reached out. No, Netflix reached out and said, "Hey, I see you've done RxJS. We do a lot of it. You want to come and interview with us?" And I was always told that you should never reject kind of a handwritten personal invitation to interview. This was way before bots and even the bots were pretty obvious to tell they were bots. This was a manager at Netflix, Jeff Wagner, first manager ever. And he just wrote a really nice note and just like, "Hey, I see you're doing a lot of these things. We really need help with JavaScript. I would love for you to come interview. We're even using a lot of RxJS if you're interested in that." And so I was like, all right, I can come and I'll interview. And lo and behold, interview went on and I called my wife I think halfway through the interview and I was just like defeated, absolutely crushed because I said... And she might remember this but I said, "We now have to make a decision. Are we actually going to move to California or not?" Because I already knew I had the job at that point. I was just knocking them out of the park. I was doing a great job on that. And so I just knew for a fact, I'm getting a job at Netflix. There's this thing that people always get so freaked out about when it comes to interviews and all that. And I luckily somehow avoided this. I don't get test anxiety, I don't get any of that because when I go into these situations, my only goal is to show the things I already know. And so it's like I walked into this situation, I've been preparing for this 80 hours a week for the last five years. So just walk in and I'm just showing the things I know. And it was perfectly fitting for Netflix at that time period in the 2013, early JavaScript days on television. And so it was just awesome, just worked out perfectly. Got hired there.

Lex Fridman

So we're in California with Netflix. This is San Francisco.

ThePrimeagen

Los Gatos. So if you're familiar, so classic symbol people do which is this is San Francisco, Oakland, San Jose, Los Gatos is just kind of a little bit south of San Jose, same mega contiguous city.

Lex Fridman

Yellowstone is in Montana, Yellowstone, the show. So is it basically like that, Kevin Costner riding on a horse? Were you riding on a horse to campus?

ThePrimeagen

No, but I love those stereotypes. Actually to be completely fair, when I was 15 years old, I was driving around on what is now a very busy populated street shooting gophers out the window of our car with a 22. So it's like Montana was a different place at one point than it is today. And there's plenty of parts of Montana that's still very rural, still kind of more of that old world. So yeah, a little bit you can get whatever you want from Montana. As far as culturally goes, I'm not really sure the best way to put the difference between California and Montana. It's just different expectations. One thing I can really appreciate about California, or at least when I say California, I mean the Silicon Valley Because obviously LA and the Silicon Valley, very different attitudes, very different mindsets. You can't really compare one to the other. One thing I can say that's really positive about the valley is that everybody is operating on this idea of trying to build or create something, and there's an energy to it that's very exciting. You meet somebody and they have a startup and they're working on the startup. And it's very exciting. And there's a lot of negative aspects to that, and we can all agree that our entire life being commercialized has probably not been that great. But the kind of the experience of being there and everyone's excited to build something, it's a really cool experience.

Lex Fridman

Yeah, it's really great. The excitement, the energy.

ThePrimeagen

Yeah, Montana doesn't have that.

Lex Fridman

I have a romantic admiration for the shows like Yellowstone being out on nature. It's beautiful. I like riding horses. Somebody also said Reddit is full of wisdom about you. Some of it could be fake news, but something about horses and this kind of thing. You like horses, you like riding horses?

ThePrimeagen

We have horses on... Our neighbor had much more hilly land and one of the horses broke its leg, so they had to put it down. And so we just said, "Hey, we are on much flatter land. You can just have your horses in our property." And so we just have horses that run around on our-

Lex Fridman

What about milking cows? Somebody asked about cattle and cow.

ThePrimeagen

So I've only had open cows. If you don't know, cow means girl, open means that, hey, they've tried to get the cow pregnant. The cow did not get pregnant first try. And so they're calling

that gene, they're getting rid of that gene. The open cow is going to now go out to pasture for the year and then get turned into delicious T-bone steaks and various things. And so we would house open cows on our property. So no, there's no milking of open cows.

Lex Fridman

Okay.

ThePrimeagen

They'd be very upset if you tried to milk an open cow because they're not milking cows. You have to get that cow pregnant. And then once you get it pregnant, you have to kind of put it into this permanent state of milking and all that. And it's a little bit more complicated than say what we did, which was just cows on eating grass and I didn't have to touch them.

Lex Fridman

Okay, well, that's wonderful.

ThePrimeagen

Reddit is not a great place for wisdom about me. They're going to give you the craziest answers.

Lex Fridman

We'll return to Reddit time and time again, my friend. So yeah, you took the leap into Netflix. So what was that like?

ThePrimeagen

This is one of those things where when you talk about it, people love to trivialize this because it's like, oh, you're taking a leap of faith by going into a fang company. And in 2013, sounds super risky. My wife was 36 weeks pregnant. We had to travel to a place where we knew not a soul. We were about to have our first kid. We didn't even have a doctor. If you don't know what having a baby does, you kind of want a relationship with a doctor. There's a whole thing that goes on there. So it was a really hard and great experience. So I went to a job in which their culture deck... So during this time, this is where Netflix still had kind of that old generation X feel to it. Their culture deck was hire fast, fire fast. It was very in your face about like, "Hey, this is how we operate. You don't meet the standards, we kick you out." So it's like I'm leaving a place where it's more secure to go to a place I don't know anybody, to a job that's bold in its claims about firing everybody with a wife that's just about to have a baby. And I'm from Montana and every Montanan's born with a natural dislike of California. So there's all these things kind of flowing into it where it's just going to be like, wow, this is a very intense experience. And it was hard for sure. It wasn't just some easy simple experience that we were just like, oh, well, I work now at Fang. We had to kind of work through that. Having a kid was very difficult. Our first kid was very difficult. Not having any family around to ever help you took a much larger toll on my wife than me, for sure.

Lex Fridman

What was the technical learning curve for you? You showed up in your plaid pants, dressed up?

ThePrimeagen

Yeah.

Lex Fridman

What did you have to learn about the Stack? Because Netflix, I imagine is this incredible infrastructure. It has to deliver just a huge amount of data. I'm just blown away by Netflix but also YouTube. These companies that have to deliver, serve a huge amount of bits.

ThePrimeagen

Netflix has it easiest. Out of all the companies Netflix by... Even though you could say maybe we beat YouTube in view hours, I'm not sure if we do, but let's just pretend Netflix has five x more view hours than YouTube. Whatever it is, Netflix has a fundamentally easier problem than all other companies. And let's get back to that. I'm going to first tell you about the Stack, but I'll tell you why it has a fundamentally easier problem. So when I first got there, they gave me my PlayStation three. My boss said, "Go learn some code. Come back to me in a couple of days and tell me what you've learned. And then I'm going to start giving you bugs to fix."

Lex Fridman

Wait, PlayStation three, what are you talking about?

ThePrimeagen

Well, I was on the TV team. I had to go plug in a PlayStation and start launching programs onto the PlayStation three and figure out how to work Netflix on a television device.

Lex Fridman

Oh, so you have different kinds of devices. Why PlayStation three, is other different-

ThePrimeagen

It's just 2013. That's what you have.

Lex Fridman

Any devices that plug into the TV? Okay, cool.

ThePrimeagen

Yeah, not as many TVs had Netflix, let alone what they called their Darwin app, which is their new application. So if you bought a VIZIO earlier that year, you'd get their older one there. It's called Plus UI. You get their older version. And so not many had the newer version. We no

longer supported Plus or we never actively developed on Plus, we only did stuff on Darwin. And so I had to learn that whole stack, the backend or the middle end, the middle layer between the actual backend and the front end was written in Groovy. And as I went around, Groovy is if you're not familiar with Jenkins, then you've probably never interacted with Groovy. But Groovy is a JVM language. It's a very interesting language, but here's how it got started at Netflix.

Lex Fridman

Oh, it's Apache. Apache Groovy is a powerful object-oriented programming language that runs on the Java virtual machine released in 2007. It has evolved to become a versatile language that combines both static and dynamic typing capabilities.

ThePrimeagen

All right, so the AI is kind of lying to you. Groovy is not a powerful great language. That statement makes it seem way cooler than it actually is. You'll meet one out of 100 people that have touched Groovy that said, "Oh yeah, Groovy's great." The other 99 will be like, "Heavens forbid, you ever have to touch that language." So when I got there, nobody, not a single soul at Netflix, those 40-some engineers had any idea how Groovy pretty much worked. Somehow people just hacked together these scripts and put them all on there and it worked. And this was before there was a Groovy RX port. We wrote our own version called WX. It was a nightmare, observables, all these things. I remember one time they told me that, "Oh yeah, with RX it's really easy. You just say what you need to do. It maps out and boom, boom, everything will run and all that." And I was like, "Oh wow, really?" So all I did was go like `observable.sleep one` because I just wanted to see it sleep and then do the next thing. And it turns out when a thread sleeps itself, no thread can wake it up. And I just turned off all of staging because I ran it like 10 times. Like, oh, it's not responding. Oh, it's not responding. Oh, now it's not even coming back. Broke all of staging for everybody. So no developer could work for the rest of the afternoon because I locked up all the instances because it turns out no, it was in fact not multi-threaded. Every assumption we've been told is a lie. No one had any idea what they were doing. It was a wild time. And so I just simply naturally gravitated towards that because I'm good at print off debugging. I'm good at doing those things. So I was like, yeah, I'll just figure this out here. I will do this. So I had the rewrite how we do the data structure on the front end for the TV from what is called a LoLoMo, list of list of movies into LoLoRoMo, which is a list of list of recommendation objects for movie. Why would we need to do that? Think about this. You have two lists, one has Live Free Die Hard, Bruce Willis because you love Bruce Willis. The other one has Live Free Die Hard because you want tough men doing tough jobs. Well, during those days we'd only have one way we could show evidence why you wanted it. So we couldn't say, "Oh, because you liked this other movie." You'd go to that one and say the same thing. So we had to add one level of indirection where we could decorate the video with the recommendation information.

Lex Fridman

Okay. So you can abstract away into the space of recommendation versus the space of movies directly.

ThePrimeagen

Yeah. So you can't hang it off the video because obviously then it would be the same for everything that shows that same video.

Lex Fridman

That's amazing.

ThePrimeagen

I had to do all this and I wrote it in Groovy and I just did it-

Lex Fridman

Such a funny name.

ThePrimeagen

And people were like, "How did you write this in Groovy?" And I was just like, "Well, I read the language reference for a day and then programmed it." Well, what do you mean? It was a very radical language, shall we say. And so I just simply became the person that knew these things, so they just give me more and more jobs with that. And so that's kind of how I excelled, being the person that was willing to do the thing that no one else was.

Lex Fridman

Yeah. Can you actually speak to the print off debugging. You walk into a system and there's a lot of systems in the world like this. Twitter was like this, when Elon acquired Twitter and the rolls in and there's this old junky code base that's just like a giant mess, and you have to basically do print off debugging. What's the process of going into a code base and figuring out what the fuck? Well, how does this work? What are the flaws? What are the assumptions? You have to reverse engineer what all these other engineers did in the past and the mess across the space of months and years, and you have to figure out how all that works in order to make improvements.

ThePrimeagen

I've always just been good at print off debugging because one of my first kind of side quest jobs that I got was writing robots for the government when I was still at school. And so I'd kind of do this contractually for so many hours a week. And my boss, Hunter Lloyd, great professor by the way, he just said, "Hey, here's your computer, here's the robot, here's how you plug it in. Here's how you run the code. Can you write the flash driver, the ethernet driver. Can you write the planetary pancake motor? Here's some manuals, I'm missing some. Just figure it out, I'll be back." So that was government work for me. So I was like, okay, I'll

figure all these things out. And I figured them all out and the only way to really get anything out of the machine was to print. And so it's like I had to become really good at printing my way through problems. And so that kind of became this skill I guess I adopted is that I can just kind of print off debug my way through a lot of these problems. Obviously I'm not a game developer, probably a different world probably should use... I think John Carmack was on here and talked to how great the debugger is, different world. Because when I was at Netflix, there's machines that exist somewhere on AWS, I'm not logged into them. I don't even know how to log into them. I'm not even sure if I have credentials to log into them. They run once somewhere and I have to figure out what happened and why it's happening. So it's like I'm going to become... This is what I've trained for. I'm a print off debugging champion. So it's just like I could just run through these things really quickly and figure out why they're happening the way they're happening.

Lex Fridman

You're a special human. I think that's an incredible skill set to have to be able to drop in into any code base, drop into any situation, and do print off debugging. Meaning you're in a dark room and you're feeling around that room to try to figure out what the room is.

ThePrimeagen

Well, I had the code so it's like I can kind of blueprint what's happening. I don't understand the services or anything, but you can start guessing pretty quick as to what's going wrong.

Lex Fridman

Right. But then the print side of that helps you confirm your intuitions, test your intuitions and build up more and more information. And then you start to accumulate this bigger picture from that, what the edge cases are that break the system and not. I think that just that kind of situation is intimidating for a lot of engineers. They break down at that point. I think it really is a powerful thing to be able to come into a code base, that's generally a skillset of very few of us start from scratch. And actually this is the fundamental problem of web development and in general where they're like, I don't know what's going on. I'm going to write my own thing from scratch. As opposed to actually doing print off debugging on the space of languages, on the space of problems, because there's a lot of wisdom and solved problems already in this code base. It's a much more important skillset to understand, to learn from the mistakes and the wisdom of the past, of the ancestors that came before and build on them as opposed to throw it all out and start from scratch. This is something obviously you see a lot with a JavaScript framework that comes out and you won every single day.

ThePrimeagen

I have a very great story about that, that this is what I think has shaped me the most about my perspective of other devs. There's this dev and he always just wrote things in just what I thought was such a bizarre and weird way, and this had to do with Falcor. So our data

fetching library for Netflix, This would run on mobile. So I had to write in Objective-C. It had to run on television and it had to also run on web. So it ran on everything. And me and one other person were responsible for this thing working. And the request side where we'd have to de-dupe the information that we already have, the requests that were pending and the new data. So I had to figure all that out based on what someone's requesting, and then just only optimally request the stuff that we don't have. He wrote in such a goofy way and I'm thinking, man, this guy is just... What a goofball. So I delete it all and I start writing and I'm like, look at how much nicer this is. It's looking so good. I'm like, Ooh, there's that one edge case. Okay, I can see why he wrote it this one way. That's not a big deal though. The rest of my code's really great. By the end of it, I'm like, I literally almost line for line just reproduced what he already wrote. It's slightly different towards my style, but I just wrote the same code. I'm like, I'm an idiot. I am the idiot in this situation because it was already a solved problem. I just didn't take the time to learn what he did. Instead, I relearned what he did by rewriting the entire thing.

Lex Fridman

I think that's a skill set that is extremely important for people to learn. I see that in myself. That's a constant struggle for myself when facing a code base, for example. But this applies generally in life, where somebody did a lot of work to do a thing, you should invest a huge amount of time and get really good at figuring out what they did, why they did it. Do a lot of print off debugging to understand what they did. It's a much more efficient way to understand a problem deeply than to start from scratch. Even though there's a constant temptation to start from scratch, because starting from scratch is fun. You do get the puzzle solved and all that kind of stuff. It's just not going to be the right thing to do. Usually pain is the right thing to do, and it is for most people painful to understand other people's code bases.

ThePrimeagen

I highly recommend starting from scratch if you want to understand a concept. You don't know how an HTTP server works, create a TCP socket, learn how to parse HTTP. It'll become very easy and you'll go, this is the reason why whenever I get a request, I have to await the text. I now understand why the text is for whatever reason not there. I get it. I now understand it. And so you kind of gain these new perspectives just by simply parsing something out.

Lex Fridman

All right. Back to the wisdom of Reddit. Apparently there are memes and legends about your programming arc in Netflix. This Falcor system you mentioned, somebody, I think it was Teej, how do you pronounce his name by the way?

ThePrimeagen

Teej.

Lex Fridman

Okay, Teej.

ThePrimeagen

TJ would be his name, but we call it Teej or Telescopic Johnson.

Lex Fridman

Oh wow, so many names. DDoS, distributed denial of service attacks, you apparently were able to accomplish the simplified version of that of just DoS. That's a legend. So you basically broke down the system somehow.

ThePrimeagen

Yeah.

Lex Fridman

Can you tell the story of that?

ThePrimeagen

I'd be glad to. So there's this Falcor business, and I did discover the bug before anybody else and I did report it to security and it was so bad. It actually got its own name, Repulsive Grizzly Attack, and they even give examples of how to do it. Effectively what it means is that there is a request that targets both memory and CPU and will destroy. There you go. Look, how Netflix... The next one down was the article that was actually written. I don't get mentioned, which is a little bit upsetting considering I was the one that discovered it and told everybody how bad it was. Anyways, and had the right to fix for it or the first fix. So this is how it works, is that you can do something pretty similar, I believe with GraphQL as well. It has the same kind of danger. Any of these kinds of RPC request as much or as little of the data as you would like frameworks, are vulnerable to this kind of attack. So with Falcor, what you do is you give it an array. That's an array is called a path, and that's the path to the data. But sometimes you don't want to have to write out, I want movie, I want row zero or list zero or row zero column zero title. I want row zero column zero description. You don't want to have to write out all that. So instead you could just be like, I want rows zero through 10, columns zero through 10, titles and descriptions. So you can write in a very compact, nice little format and it'll give you all that data. It'll go to the server. The server will fill that all in and give it to you. Oh, dang it, list three, it only had three videos in it. So what happens when I try- Three, it only had three videos in it. So what happens when I try to re-request the data? Well, I need a way to be able to tell my system that you'd have requested the data and there's nothing there. So call this like a boxed value. So it's going to be like type, something, value, there's nothing there. We've already requested it and there's nothing there. It's like a sentinel value, if you will, a boxed value. And we have this little special flag weed pass called materialize. Meaning that when you ask for a path, we will make sure we fill it out so we don't accidentally erase anything. And at the very end we'll say okay, the thing does, the request

you've made has already been made and there's nothing there. Well, what happens if I request rows zero through 10,000 columns through 10,000, one more item through 10,000 and then a whole bunch of properties and then ask it to materialize? Well, I'm about to go create billions of objects in the JVM, and what happens to the machine? It stops running. And then if we try to JSON... Even if it could create a mall, we then ask it that JSON serialize, it's not going to do it. It's impossible. And so that was the attack vector, is a simple wild loop would've taken down and held down Netflix for a very long time. Because one request would kill one machine on AWS. And so that means it would just turn it all off. And this was on the website? This was on TV, this was on mobile. This was profound. And here's the worst part, it was in production for years so we couldn't even roll it back. There was no like, "Oh crap, let's just roll back to two weeks ago and we'll fix forward and figure out." No, it's like we could roll back to 2011. That's our option is 2011 and that's it. So we had to figure out a way forward and all that. And so it was like... The amount of problems that would've happened if someone would've discovered this is unstatable.

Lex Fridman

Just to be clear, the infrastructure that's serving the videos would shut down.

ThePrimeagen

Yeah, the UI, you couldn't perform any actions in the UI. You surprisingly could still stream video but you would never be able to get to a video to stream. Because every action you would take would be completely shut down. And so it wasn't a DDoS because you didn't need a bunch of computers to try to overwhelm the system by making a bunch of requests, one request, one machine. If we had 50 machines serving the millions of requests, it'd only take 50 requests to shut down the entire UI.

Lex Fridman

Isn't it possible to do DoS or DDoS on basically any software system Like defending against all the closing all those attack factors is probably really difficult. If you take any sufficiently complicated software system, there's probably so many ways to overwhelm it.

ThePrimeagen

Yeah, this is why people use CloudFlare. I think d HH said it best, which is like we have our website and we have a strong bodyguard on the outside. So CloudFlare has a bunch of utilities all built in. Because obviously this is why everyone hates all these Bluetooth devices that connect to the internet because they just turn into attack vectors where people use those to DoS or DDoS other sites. And so you don't need something sophisticated, you just need a bunch of requests to come in and you can take down websites. And so that's why these fronts are really good at discovering where these problems are. But DoS is a bit different, because it doesn't have to be overwhelming by using resources with a whole bunch of requests. It really just means simply that there's a denial of service attack. One of them could be there's a RegEx attack that existed where CloudFlare actually did it to itself

and shut itself down, which is there's a RegEx expansion attack where given the right RegEx, if you know someone's running a specific RegEx, you can actually provide input that is maximally bad and that thing goes to super processing. It takes 10 seconds to process a single request, then you only need to make hundreds of requests and you shut down the whole service. It's not like you need some giant machinery to make one trillion requests. You only need just some small amount to completely destroy a service. And so there's... The web is an extremely difficult place to do it correct.

Lex Fridman

This is super fascinating. I do also wonder how many ultra competent, what is it, black hat hackers there are, versus the good guys versus the bad guys. How many bad guys there are and what is the average... What is the distribution of skillset on the bad guy side that are constantly trying to attack?

ThePrimeagen

I assume there's probably a huge number of just really simple ones, script kitties, right? Just people trying to just do things. And then there's a huge amount of social engineering that just goes in where hacking's done, not with a computer but just by one of the classic ones. Kevin Mitnick had this one in his book which was you'd call up somebody pretending to be like, "Charlene, we're doing some auditing and I think your pin's out of date on file. Is it 2323, still?" And they're like, "No, it's 4747." You're like, "Oh, thanks Sharon." Boom. You just hacked him. Right? The classic people love correcting bad information. This is like a standard. So there's all these ways people hack. And so my assumption is that there are really great white hat hackers, there's really great black hat hackers. But the vulnerability space, the harp, the thing is that discovering a vulnerability and you don't let anyone know, the white hat hacker still has to make that same discovery. And that's where I think the real thing is that black hat hacking in some sense has a fundamentally easier job or at least a job in which they can take advantage of for much longer periods of time. One's the process of discovering who's breaking the system. The other one's trying to figure out how to break the system. And it seems like most software is held together by toothpicks and glue and there is a lot of dangers in every piece.

Lex Fridman

And also the social engineering aspect, that's a real attack vector. I think that's the attack vector that will do in the longterm the most damage in the world. Especially as AI tooling becomes easier and easier to convince people at scale, sort of do that email Grandma. I think that's a really serious attack vector, like human psychology and all that. I assume whenever there's a girl that approaches me, it's some kind of social engineering project, some attack vector, some intelligence agency. In fact, I'm pretty sure-

ThePrimeagen

We're back to A Beautiful Mind, aren't we?

Lex Fridman

Beautiful mind? Yeah. I have a whiteboard upstairs that I calculate everything, everybody's trajectory and move.

ThePrimeagen

You're not wrong though with the attack vector, especially in the day of AI. One thing that I don't think a lot of people are talking about as we integrate more and more AI is that prompt injection is an extremely hard thing to defend against because it's not really clear how you defend against it. If it's just a, at the end of the day word calculator make word come out. If you can figure out the proper word calculator input, it might just break its bounds and start doing something it's not supposed to do. And there's a whole future word. There's all these products that are going to be vulnerable to things they never thought about. It's one thing where you forget an edge case while you're programming. Now you have to guess what people might be able to think of making something that has access to a system be able to do. Right. And you don't have a way to reason about it. Its reasoning came from Reddit, and other words that it's read and how to put things together. This is a very... It's a massive space that's going to be happening. It's why I'm personally thinking don't give too many powers yet. We don't know the attacks that are about to happen.

Lex Fridman

Yeah, the more power we give to software systems, the more damage they can do. That certainly is the case. But the more awesome they could do, and that's the knife's edge that we all walk along as a human civilization together, hand in hand. Will we flourish or destroy ourselves? Question mark. Folks on Reddit, the good folks on Reddit, demanded that I ask you about the time you broke production. Is this related to Falco? Did you break production? Is this fake news?

ThePrimeagen

I've broke production quite a few times. I've broken productions for so many stupid reasons. One time I broke production because I came up in the PHP and PHP. Static means static for the lifetime of the PHP and PHP was the lifetime of every request, right? That's why PHP was so inefficient was that every request was its own instance, and therefore static memory was for the lifetime. I guess I never put that together. And so I had some objects that I made static because I was like, oh I just need this for the lifetime of the request. And lo and behold, those weren't lifetime. A whole bunch of bad data got all over the place. People were showing up saying they were from all these different countries and everything was all wrong because I just... "Whoopsie-daisies." I just made a whole conundrum with that. So that was one time I did it. Another time is I took down, if you were on the homepage on the website waiting for Lady Gaga's video to come out and you were watching the countdown go down, if it reached zero, the billboard would freeze and it wouldn't work. If you refreshed it, it would work. But the reveal, the big reveal, I screwed that up and my boss got real upset and so did other people in Hollywood got upset about that one. That was like

a, "My bad. Sorry, Jeff Wagner, again." I remember that one. I remember that one specifically. One time I released a bug where again on the billboard, if you pressed add to my list, I accidentally programmed in an infinite loop, and your whole webpage would just freeze.

Lex Fridman

Are some of these bugs difficult to discover until you started-

ThePrimeagen

That one seems really easy looking back at it.

Lex Fridman

In for a loop? Yeah.

ThePrimeagen

And we actually, during those days we had manual QA that are supposed to go through everything. So I didn't feel as bad because my manual QA counterpart also missed it. We all missed it. But it was just so simple. Just press that button, boom. It just completely freezes the website.

Lex Fridman

Polluting the code with global variables that are holding values, SPHP I think allows you to do, that's a tricky one to discover, because you rely on it, then there could be somebody else assigns a value to it.

ThePrimeagen

Yeah, it's a data race everywhere. And I just didn't understand... In my head static was like, "Oh, this is for the life." I was just so locked into the PHP world at that time that I just made just such a, looking back on it's so obvious. But during the time, it's hard.

Lex Fridman

So in general, pushing to production, I talked to Peter levels about this. He, obvious he's operating as mostly a solo developer, but he often on the website said thousands, not hundreds of thousands of people use. He often ships to production, pushes to production, meaning just no testing, just like push to fix. What are the pros and cons of that approach in general to you? What do you think?

ThePrimeagen

It's obviously much easier the smaller your organization is. I think no one would argue that sentiment. If it's just you working on a singular project, it is obviously much easier for you to push directly to production because you are the only one working. You know all the ins and outs and if something were to break, you would discover it. So to me that makes sense. I

think the way he operates is perfect for what he does. You couldn't take what he does and move it to say Microsoft or Netflix or Google because that would obviously... It would just be a disaster, just due to the amount of people all pushing to production. And so I personally love that. I think that you have to gauge both the application you're building and its complexity and what you're pushing, and how many people are working on it. I think those all go into how you can do that. Because not all applications are created equal either. That application I was making was zooming and scrolling where we had all of our own everything. It was a very deep heavy logic app, and that was regardless of what was happening on the website, most of the code was library code. And that becomes way harder if you don't have a good test suite and stuff to run before you push it out. Because when you squeeze that ball, different things come popping out in different areas. And that's a very harder problem than say if you're doing more of a heavy visual one because a heavy visual one, you're affecting just this one area's visual stuff and you can test it and that's normally the end of it. Whereas, you know... So it depends on the coupling and everything. So I love his approach by the way. I have such mad respect for anyone that operates that way because I think is a great way, it just is so good because it breaks this notion that tech Twitter has that oh well you have to use all these expensive services, you need to use all these things because if you don't use all this stuff, "If you're not using the latest version of React, if you're not using the latest version of this, you're simply not going to make it as a startup. It's impossible." And it's just like, "No, no, that's not software." Most of software isn't the new stuff. Most of software is old crappy software that someone has to maintain, and it actually is really, really great and has lots of really hard problems. And if you look at it differently, it's actually fantastic.

Lex Fridman

For people who don't know his tech stack, in terms of web development is PHP, jQuery and SQL.

ThePrimeagen

Yeah, all great stuff. I'm just surprised he still uses jQuery just given the fact that at this point on the modern web, everything is, you have document query selector and an event listener click, right? It pretty much has everything you already need. It had DOM content load, all the reasons I used jQuery back in the day was adding a click on a button was hard. You had to deal with IE7, IE8, IE9. Those are hard differences. Whereas now, it's just so easy. I'm just surprised it's even that.

Lex Fridman

That's definitely a trade-off. I still use the exact same stack, PHP, jQuery and different flavors of SQL. But the question there is you keep using jQuery because you can get the job done really fast and there's no significant performance hit that you detect. So like why switch to something else? But it's always probably as we'll talk about good to explore and to learn.

ThePrimeagen

Not all tools are great at solving all problems. And so what you think is really the problem is you run into this trade off, which is you have some tool belt that you're very adept with, you know all the ins and outs. There's no unknown unknowns, but there's no surprises in this. You know what you're building, you know what you're getting into. You will go through and you'll be able to solve the problem. But if you ever use a different language or a different experience, you can find that some things are able to represent states way easier, in a way more efficient way. And you can solve problems really efficiently in some versus the other. And so it's like if you don't take the time to explore as well, you could be missing out on something that makes you twice as good on this one specific problem like subset. And so I value being able to look at all problems. And so I don't want to get stuck on one thing though. I see why people do, which is for the efficiency sake.

Lex Fridman

Let's just return to the infrastructure of the platform of Netflix and, speak more generally, Netflix, Twitch, YouTube. Anytime I use any of these services, I'm just blown away by the infrastructure it takes to deliver this service. YouTube and Twitch are unique, versus Netflix where the creators can roll in themselves and upload stuff. So on the consumption side, YouTube has over 100 billion views a day, over one billion hours watch time. But on the creator side, one million hours of videos are uploaded every day. One million hours. It's like you have to service both and you have to deliver everything... It's just incredible to me. Can you maybe speak to your own intuition, just zooming out on it, what it takes to deliver that kind of infrastructure?

ThePrimeagen

For me, the thing that I find vastly complicated and I can't imagine the engineering hours, is how do you even create an edge in that situation? And what I mean by an edge, when people say this phrase, if you're unexperienced, an edge is where you deliver data. You want that edge to be as close to the customer as possible because that's where the data lives. And then the communication between the customer and what you're doing is really, really small. Obviously the speed of light adds up, the amount of hops adds up, the amount of services that you have to remotely call adds up. They all add up and they all add inefficiencies to the system. So something like YouTube, they want to be able to serve that data as quick as possible, but their data changes constantly and relevance is almost directly tied with the newness of the item. So it's like how do you even cache these things out? How are you doing this? So they must have such an incredible caching network that I can't even... I can't even fathom what it takes to do that. That just to me is just so impressive. A million view hours in how many different resolutions with how much data? What is a million view hours? Is it 4K million view hours, along with 1080p, along with 720p, along with 1440p? That number is an insane number.

Lex Fridman

Actually, it is brilliant what you said, which is for YouTube often the new thing is extremely important to show to everybody. And so, you can't rely on caching or trivial kind of caching.

ThePrimeagen

Yeah.

Lex Fridman

You have to deliver the new thing as quickly as possible. Yeah, it's incredible. So there's the entire system, the recommendation system that knows each individual human watching YouTube and it has to integrate into that the new thing, while also caching this incredible cluster of possible videos that you're potentially interested in. And integrate into that ads in the case of YouTube and so on.

ThePrimeagen

It's a really tough problem because you have to think what is the cash hit rate on this? Because the problem now actually comes down to space, space actually becomes a real problem. How many hundreds of petabytes do they have that they have for like, "Okay, what do we cache and where do we cache this?" The number, I think in the terms of gigabytes or maybe megabytes, they have to think in probably versions of bytes I don't even know the name for right? It's such a different problem and that's why I said Netflix. Netflix has a much easier job when it comes to caching. So if you've never looked it up, it's called OCA and that we know what videos we're releasing, we know what videos are hot in specific areas. It's a very limited set. We're not going to all of a sudden get, "Oopsies, we got a million new view hours." We don't even have to worry about that as a problem. So it's like, "Okay, we know Stranger Things season five's about to drop, we're going to pre-cache Stranger Things season five in every single OCA across the world because that thing's about to get hammered." And so it's like it's able to do such a different decision-making than what you have to do with something like YouTube. And then Twitch is even more wild because now you're actually ingesting video and trying to make it go out all at the exact same time for all video and you have to transform that video from whatever format and whatever the bit rate is into something that's more efficient in the system like that. Hats off to Twitch engineering, because that's some serious work.

Lex Fridman

And here's some asshole, Lex, coming out and tweeting about YouTube features. Listen...

ThePrimeagen

You're not wrong on the features you asked for, though.

Lex Fridman

I think this is an engineering problem of how do you allow fast iteration and addition of features that shouldn't have to be integrated or impact the whole code base. So at the edges of the code base improve on certain features, without having to consult the mothership of the code.

ThePrimeagen

It's the large team, right? That's the fundamental problem. When you get into YouTube size, there is the team/organization that deals with data warehousing. There's the team slash organization that deals with delivery. There's a team slash organization that's like the middle layer, how you even... They're going to be like the little micro-surfaces to talk to these places. Then you have this front-end engine. So for a small feature, you have to get middle team, you have to get backend team, you have to get all these things. Quick example, Netflix. Are you familiar with the dystopian, Black Mirror?

Lex Fridman

Yeah.

ThePrimeagen

Okay. Season one, episode one. Do you know season one, episode one? Everyone who watches Black Mirror typically knows this episode.

Lex Fridman

Okay, yeah. I don't remember what it is.

ThePrimeagen

Forgive my language, but they call it the pig-fucker episode.

Lex Fridman

Oh yeah, of course.

ThePrimeagen

Once you've seen the episode, you will then know this episode. Well, when Netflix adopted it, I got pulled into a room, there's like a VP, a product designer, a VP, and they said, Hey, we're about to release our own version of Black Mirror, season three, I think at that time. We need episode one, season one to not be the first thing people see. So let's just reverse the season order. That required me... I had like 20 engineers I had to gather together to be able to have this happen. And that's just the problem of big companies is that eventually every little thing has to become its own team. And so even small... There's no such thing as a small feature.

Lex Fridman

Reversing the order of the dropdown that selects the seasons is a meeting with a bunch of VPs and engineers. That's really interesting.

ThePrimeagen

Yeah.

Lex Fridman

There's got to be a way to accelerate that. The natural scaling of a company and the bureaucracy that grows, yes, slows that down. But just having seen Elon work a lot, his teams are able to still keep it very fast, even as the company grows. There's got to be a process to doing that, especially for the pig-fucker episode. I don't know where that's in the priority list, but for important things like that, you should be able to do that quickly. I don't know. Can you speak to how would you do that?

ThePrimeagen

Well, I can tell first how it was done. Remember... So at a place like Netflix, there would be... I think that at that point it's called a product called Dexter. I can't remember. There's our actual movie metadata warehouse that's going to be highly integrated with Hollywood, where that side is able to manage all that. So I'm like, "Hey, you need the ability to mark things that need to be reversed because we're going to run into this a bunch." And we did. We ran into quite a few topical shows that all need to be reversed and all that. And so it's like, "We need to be able to reverse episode numbers, season numbers. We need to be able to hide season or episode numbers." In the case of the Chelsea Handler Show, it was like a daily show, so it's like you don't need episode numbers, you just need the latest one. And so there's this whole problem that exists. And so it's like, "Okay, you need to work on that for your UI over there, then you need to be able to store that data. Then we need to be able to go to the people that can actually get the video data out of that and provide it to our service layer. I need to go talk to them and convince them they need to be able to give me the new methods and everything to do that. Then I need to be able to go write the methods to get it down, and then I need to go to the UI and make that accessible. Now I need to go to the website people, I need to go to the mobile people, I need to go to the TV people."

Lex Fridman

Yeah.

ThePrimeagen

And so it's like, you could see this thing like snowballing. And for us, the big thing that Netflix did that was so well is after I met with these people that were high level, I was the captain. "I'm the captain now." So I went to all these teams and said, "Hey manager, I need an engineer. We need to get this done within the next couple of months because we got Black Mirror coming out." So she would go, "Okay, here you go." "The map team, I need someone to

help me with being able to get data out of the lomo for this." And so it's like, "All right, you're working with this engineering." I'd go to the BMS team, "Okay, I need this engineer." I'd go to the billboard team, "I need this engineer." I go to all these little places to get all these little pieces of data. And then I was the captain, I was like, "You're working on this, you're doing this, you're doing this, you're doing this, I'm doing this. Let's go." It's like that worked. And we were able to go pretty fast for a big company. And the fact that it required 20 engineers to do such a simple task, we were able to do it in, gosh, I'd say about three weeks worth of effort. But that was still... I thought that was amazing, comparatively, to how many people move.

Lex Fridman

Well, because you have the freedom of the agency to do it. You said the captain of the ship. That's really powerful. For big companies, that's a risk. Because you can fuck it up. You might not see the bigger context legally the bigger context of the impact on the industry or all the contracts that are made, all that. So it's a risk, it's a risk, but it's a risk you have to keep taking. And then when you fuck up, you fix, and then maybe pay the cost legally for that, whatever. But the long term, that risk pays off because you're going to keep creating a better and better product, evolving where the industry is going, constantly innovating ahead of where the industry is going and so on. Yeah.

ThePrimeagen

And not only that, I think one thing that is just so important is that yes, the product will get better, but the people that you hire and the people that you keep around are better because they're the ones that show maturity. They're the ones that can just... You give them something and they can rally the troops and make something happen. That's a very great group of people to hire. And so you also naturally select out great engineers that aren't just simply good at coding, they're good at coding and they're good at explaining and they're good at convincing. And you have to create a very lean audience that can move fast.

Lex Fridman

And I think for great engineers, having to wait for like, "Okay, let's schedule a meeting for next Wednesday with the VPs and..." That destroys their soul. And they either don't want to contribute anymore or they leave the companies or they just tune out and take the golden handcuffs and just buy a nice house and focus on a family.

ThePrimeagen

Yeah. I feel like I would die under that... Honestly, that is my death sentence is where it's just that there's no reason to try, there's no reason to do anything. I'm just going to go in there, effectively zombie through my day and call it... I don't want to live like that. I want to feel like I'm trying to do something.

Lex Fridman

I should also mention on top of that, so you've brilliantly laid out how incredible the challenge that Netflix has to solve. On top of that with YouTube, the metadata thing, because users are able to upload video and there's an API where they can upload automatically and change all this kind of stuff automatically. Every one of those things is an attack vector, as we mentioned. That's something they have to consider seriously on the engineering side. And on the legal side, they can get into trouble all kinds of ways. So they have to consider all of that, which is fascinating.

ThePrimeagen

The legal side is obvious, but it's not really like... I would never have initially thought someone would, say, upload images that you're not allowed to own or have. But that guarantee you that happens. Then you have the whole kid side, right? I think about when you mark something as kid-friendly, how many times have they snuck porn into a Taylor Swift video or whatever? That was like a few years back, there was that whole Taylor Swift or whatever. I forget what it was, I thought it was Taylor Swift. But there'd be these mock videos that'd come up and then, boom. It's like, that is such an awful problem and I'm so happy that is not a problem. I have to try to figure out.

Lex Fridman

Okay. So yes, YouTube and Twitch and Netflix are doing an incredible job. You eventually chose, the madman you are, to leave Netflix and to start on a new journey of being a wolf pack of one, start streaming. What was that? What was the story of that?

ThePrimeagen

So I was streaming for almost seven years now. It started actually at Netflix. We did a charity, Extra Life, shout out to Extra Life for starting my streaming career, effectively is just you stream and whatever money you raise, it goes to Kids with Cancer research. They are a great charity in the sense that they take no overhead and they raise their own donations for their website and everything. And so it's a very great, straightforward charity. Really love what they've done. It was super cool because I live in South Dakota now, but I actually could choose a hospital directly where the money goes to. So there's a direct impact from A to B. So it's a pretty cool organization. And so my friend, Guy Cirino, Nice Try Guy is what I like to call him, he was probably the single greatest engineer I've ever met in my lifetime. And he was just like, "Hey, come do this. We're going to all do this." So I played Fortnite. So before I did that, I was like, oh, I better learn how to stream first. I better get affiliated so I can take subscriptions. And then if anyone gives me a subscription, I'll also pay that forward. So June 2018 or something like that, I start, I start streaming and I start streaming some Fortnite. End up getting affiliated, end up doing the whole extra life thing. I end up really enjoying it. I'm like, "This is a lot of fun." I'm playing Fortnite at that point. Okay, so mind you, I'm a Fortnite streamer at that point, and I start really enjoying it. I keep doing it and then one day I decide I'm going to do some programming. Because I really love Vim

and I think I'm fast at Vim, and maybe people think programming is cool. Because there was no really programming section at that point. And I did it. I had like 30 people show up, which was just... And it felt like incredible numbers at that point. So I was like, "Oh my gosh, there's like 30 people watching me program." So it just kept on going and it kept on happening and it just kept on growing. And I did it for year after year. I would do my job, I would come home, I'd eat dinner with the kiddos, I would read them Lord of the Rings and the Hobbit during that time, I'd read to them for a half an hour, then I'd set that down. And then three nights a week I would program until like 2:00 in the morning or play video games until 2:00 in the morning streaming and building up this whole side thing. And I did this for a long, long time, and then eventually it just kept working out so well and I started making YouTube videos. And then that started getting better. And it was just a long, long grind until April of last year. I went to the Streamer Awards and I got to announce the programming category and Pirate SoftwareOne. It was awesome. It was a great time. And during that time he gave me a challenge coin and just said, "You just got to go for it. Just go full time." And so I just sat there and my wife can attest to it. It was like an emotional turmoil thing and it just took a lot of, it was pretty awful because Netflix is very safe option. It was both very fun. It was challenging. I liked a lot of the people worked with, it was overall a really great thing. I had a really great boss, really appreciated him. I still have text him now and then he's really great guy. So it's just like I'm leaving all these things for something that's unsure. And the reality is that streaming and all these things, people love you one day, they could hate you the next day. There's all this stuff that goes into being on the public side. And I had Netflix as the backing, so it's like if public hated me the next day I'd be like, deuces, I'm out. I don't care now it's like, now I'm going to do this as a job. And so there's a whole huge turmoil to this whole thing that went through it. And eventually I just said, okay, I'm going to make this. It resonated with me when I first made the decision to join Netflix. I'm getting older. There's not a lot of chances to do something unusual. Those chances go down constantly. As you get older, this might be the last crazy thing I get to do. Let's just try it. So in April I went full-time and I guess I haven't looked back. I'm only not even a year into doing this as a full-time gig. And it's just been a lot of fun. And the biggest thing is just being able to really explore and do these things on stream where people really enjoy watching and engaging. It's been a great, hard, fun, amazing, difficult experience.

Lex Fridman

It's a really inspiring leap. It's a really hard one to take for many reasons, like you outlined, but also the loneliness of it, I think it's a pretty lonely pursuit.

ThePrimeagen

It is.

Lex Fridman

Just you and the camera and the audience and the ups and downs of that. And there's not really a team.

ThePrimeagen

I do have one lucky thing I'd say that, my editor, Flip, shout out Flip.

Lex Fridman

Flip, shout out.

ThePrimeagen

He said it would mean the world to him if I said, "Shout out, Flip."

Lex Fridman

I love you, Flip. We all love you.

ThePrimeagen

Oh, man. He had, as he would say, he had nothing going for him. He had a really hard growing up. A lot of rough life decisions have gone into his life and he's crawling back out of it. And he just said, Hey, I'll edit full-time for you. So I just said, "All right, like 50/50, whatever I make on YouTube, you get. We're going to do this together." And we did that for years, making \$0 a month pretty much. And so it's just like that was an incredible jump and now we get to work together. So I do get that one team aspect that I think is really nice. But it's not like it was at Netflix where I could hear about stuff people are building, I don't have a team, I don't have product or cycles, I don't have a manager that I have to try to make happy. It's just like... It is very lonely. And I don't think a lot of people realize how lonely it actually can be.

Lex Fridman

Yeah. So combine that loneliness with, in my case, I don't know how many people attack you.

ThePrimeagen

I have a shockingly low amount of attack rate, I feel like.

Lex Fridman

Yeah, people generally... It's sometimes fun sort of teasing, that kind of thing, but it's mostly just really... You give so much love to the world and inspire so many people, even when you're making fun of stuff, yeah. But with me taking the loneliness of it combined with just really intense attacks, it's tough. It can be rough. Psychologically, really a tough journey. You miss working with a team, just from even a software engineering side, where you can share code or talk over code?

ThePrimeagen

Yeah.

Lex Fridman

Yeah, the collab... Or talk over code or the collaborative aspect of it.

ThePrimeagen

Yeah, multiple things there. One, hey, we love you Lex, so don't let the things get you down.

Lex Fridman

Thank you. Thank you. I love you too.

ThePrimeagen

Thank you. Hey, little bonding moment here going on. But one thing I really miss-

Lex Fridman

Not in a sexual way, just to be clear.

ThePrimeagen

The tension is a little tense.

Lex Fridman

I'm getting uncomfortable. Yeah. Anyway, team.

ThePrimeagen

It's just the one thing I really miss is just, even when I hated how people did it, just seeing how other people solved things, it's really amazing just the raw creative power so many people have, and just being like, oh, wow, I would've never done it this way. Crazy, right? Wow, this is awesome. And then you kind of internally process this and you're like, oh, I now have a new little tool in my tool belt. Because at some point it's really hard to find a mentor when you're first, young and you're just starting out programming. I mean, anyone with a couple years of experience will be not just a little bit better than you, but infinitely better than you. It feels like crazy how much better people are. And so you have to get mentors and you learn from people. And then as you get better, that amount of availability gets really small. And so it's something that I really do miss is that forced hard problem solving together.

Lex Fridman

I think there's also a skill to mining the wisdom from other people. I generally try to approach even junior people, young folks. It's just mentally, at least for me, it works as a hack to assume they're the smartest person in the world, way smarter than me. And so I take every single word they say as potential wisdom, and that helps me sort of mine for potential wisdom there. Because it's so easy once you get older to judge, to be like, yeah, okay, okay. I've been through that. I remember feeling like that. I remember thinking that. That's incorrect, whatever. But just kind of assume that I don't know what the fuck I'm doing,

and the other person is this sage. And in that kind of interaction, I think you could actually learn a lot. And my favorite interactions is when we both think that way. So from there, I think that's a catalyst for a great collaboration and interaction.

ThePrimeagen

It just also makes everything much nicer. It really stinks to work with someone that's combative and negative. I don't mind combativeness if it's like I'm trying to figure out what's best to do right now, versus combativeness just because you're a negative person and things have to be this one particular way, because if they're not this one particular way, it's the end of the world. And that's actually really hard for me to work with.

Lex Fridman

What's the origin story of ThePrimeagen name?

ThePrimeagen

The origin story of ThePrimeagen name was, are you familiar with a video game called Turok? Nintendo 64. Turok had Turok I and then Turok II. Turok II was a brutally hard game. This is back when first-person shooters, they would only give you a certain amount of health, and you had to go discover health and get that health. And you had to beat the whole game without effectively dying. That's the first version right there. That's like Turok I and Turok II.

Lex Fridman

Turok is a renowned first-person shooter video game series featuring dinosaurs, action, and sci-fi elements. The franchise has evolved significantly since its inception in 1997.

ThePrimeagen

There you go. So in 1998, there, you can see it right there.

Lex Fridman

Turok II, Seed of Evil followed in 1998 featuring larger levels, more challenging puzzles, and deadlier enemies.

ThePrimeagen

The notable difficulty, it was very, very, very difficult. So I spent, when I got it, it came in a black cartridge, not like your standard gray Nintendo 64, the black cartridge. Badass game. And I got it and I put it in and I played, and I played every day for 10 hours a day, for a month straight. And I beat it. And it was such an incredible, great experience. And the last leader of Turok II is called the Primeagen. And so when I was a kid, when you're in fifth grade, that's super cool, named after the bad guy. And so for a long time on any internet thing, like Grail Online that I mentioned earlier, the name was ThePrimeagen. It was great. And then I became an adult eventually, and it's just like, okay, I'm an adult. My name's Michael Paulson

underscore. And that's what I was on the internet for a long time was that. And I remember it was like 2017, 2018, somewhere in there. I remember just how bad the tech world had kind of become. It was just like this super pretentious place, tons of dick measuring, just everything that just was the worst. Ken Wheeler got canceled over playing the Circle game. It was just like, it is so hard to describe to people that weren't there, but it was just the worst place to be. Tech was extremely unfun. It was extremely awful. Everything was just so, it wasn't academic because it was research. It was like we're building the most sophisticated things, and this is for the smart people and everyone else is the dumb people. Don't worry. We'll design for you, dummy. We'll show you how to make the perfect architecture. And I remember changing my Twitter handle because I got so upset and just went back to my video game name. I was like, I want things to be fun. I want this to stop. And so when I started streaming tech, my goal became to destroy whatever that tech mentality was, because it includes nobody. Everyone thinks that they're the smart people and they design for the dummies. And it's just like, no, I want tech to be this place where people feel like they can be creative, and excited, and actually build something. And if you're new, it's okay to be dumb and ask dumb questions. Learn from your dumbness. No one's expecting you to be smart. Pick whatever you want. Actually do something and have fun and build your crazy ideas. Oh, you're going to reinvent the wheel, reinvent the wheel, understand what you're doing, learn it really good, and interact and stuff. And it was just so different than what was out there. And the name... Arnold Schwarzenegger talks about this thing where, when he first started acting, his name was the thing that people hated. As he once said, you have a strange voice, you have a strange body, and your name, your name's unpronounceable. No one's going to Schneitzinfinitzel, no one's going to remember that. And he said, but now the name is the strong part. And for me, I've always felt akin to that, though my name's not nearly as cool, nor am I as popular as Arnold, nor am I as tough or good-looking or successful. But nonetheless, it's just the name represented this counterculture movement within myself, in which I just hated what was there and I wanted to defeat it. And so this has been the thing. And now people remember me so well because of how weird my name is. And so it's just like for whatever reason, it became its own thing. And so that's the... Now I would never change it, and back then I would never change it because it was my rage against the machine moment, if you will.

Lex Fridman

Yeah, I love that as a symbol of rage against the machine and the rage being fun.

ThePrimeagen

Yeah. I just want people to be creative and have fun again. It's okay.

Lex Fridman

What about the mustache? It's an epic mustache. It's an epic stash. It has a life of its own. Is there an origin story or did you guys discover each other at some point? Or did it emerge from the darkness of the struggle that is your life, or where does it come from?

ThePrimeagen

Well, the original mustache is that it was no-shave November back before it became Movember. It was no-shave November back in the day. And after no-shave November, you had all this hair. And so what's the natural thing you got to do? You got to sport a mustache for a day. So whenever I'd forget to not shave for a long time, and then I'd let it start growing out really big, I just go, oh, this is kind of funny. I'll have a mustache. So one day when I was streaming, it's just one of those times I just didn't shave, and then I started just letting it go, and then I got kind of a beard, and then I just had a mustache. When I did it, people were just like, yeah, it's mustache time. And I was just like, heck yeah, it feels like a lifestyle decision. This is the fun times. And so all of a sudden it was just exciting to have a mustache. And I shaved it off and I was like, oh, okay. But then part of me is like there's this weird energy that comes from just having a mustache. So I was like, I'm going back. Told my wife, forgive her. She was very not as thrilled about my decisions to have a mustache long-term, but I just decided to have it back and it was the right thing. It's always been the energy that I had was the mustache. It was always been there. It just never was visible until later on it feels like.

Lex Fridman

Yeah, we're chatting offline how one of the components of a successful relationship is sacrifice and your wife was willing to take the sacrifice of allowing you to have a mustache.

ThePrimeagen

I clearly was not willing to sacrifice not having one.

Lex Fridman

You do this incredible thing where you tried a bunch of different programming languages when you stream. You go all out on certain programming languages like Rust and then go and then try to pick a new one, but also are experimenting constantly. So maybe one question I could ask is about learning. What's your approach to learning a new programming language, and maybe what's your advice on learning a new programming language when you begin that journey?

ThePrimeagen

So I've kind of done a bunch of different ways to go through this learning process, and I've tried a lot of different ones. Something that is obviously successful is just start building something. Just put your hands on the keyboard, especially if you already know how to program. You're like, okay, I'm now using Zig. How do I do a main function so I can just run the program? Okay, now I know how to build. Okay, how do I do an if statement? What does it look like? Okay, how do I declare my own functions? How do I do modules, right? You just kind of Google your way through it, if you will, to get to the end product and build something. It's a great way to do things because I find that repetition, rote learning is obviously the best way to do this. You have to kind of go over it a bunch and you can definitely get out and build a lot of stuff with that. I like that initial kind of get used to things. But on top of it, I find that,

by doing that, you also fall into traps. You kind of Google and you try to solve a problem in the language based on all of your previous experience. And so you don't have what makes that language special. You have what all the other languages make special. And so you end up not really being able to use it very effectively, but you can certainly kind of learn it and get kind of good at it. And so the second approach I've been doing lately, and this has been inspired by the creator of Ghosty, Mitchell Hashimoto, is to just start by reading the language reference, the whole thing. And so lately I've been just kind of going through and just reading the entire manual for these languages. Like Zig, I'm almost done with that one. It's like eight to 10 hours of just sitting down reading, and I'll whip out my computer and kind of practice a couple of the things from the actual docs, and that way I can learn all the things. So then when I start building again, I'll remember, okay, I know there's a thing over here, let me go reread about it because now I have it indexed in my brain somewhere that will remember. And so I don't think there's a right or wrong way. I mean at the end of the day, the right way is always that you have to build something eventually. You cannot just read about it. You have to put your hands on the keyboard, you have to build something out. And then once you do that, that's where you really discover what makes it painful or what makes it great. And if you don't have the breadth of what the language offers, you just may make it painful by simply being bad at it.

Lex Fridman

Where exactly are you reading this-

ThePrimeagen

Language reference.

Lex Fridman

The language reference.

ThePrimeagen

So it just goes through every feature top to bottom, right?

Lex Fridman

That's a lot. Yeah.

ThePrimeagen

Every way it's described, all the different things. I think Zig's is, it's a decent size, but it's not just simply read the words. You want to internalize each concept as well. So it takes a long time. So I'm a slow reader.

Lex Fridman

So you're building, in AI terms, like a background model. I don't think you can just start building once you're done reading because you probably forgot how to do a for loop. You

kind of forget the specifics. You just are building up the design choices, the set of features available, what are the strengths and weaknesses, all that kind of stuff. And then you start building. That's really interesting. Probably not the thing you would recommend to a junior developer, somebody who's just starting out at first.

ThePrimeagen

If you don't know what an if statement is, that's not a good way to learn. To me, the best way to learn that is really hands on the keyboard and building extremely simple things, and slowly growing in complexity. Because understanding what a class and methods and instances versus the blueprint, which is the class versus functions versus modules versus all that stuff. That just takes time to learn. And so that's a completely different style of learning.

Lex Fridman

I wonder because for me, learning right now, AI is a huge help, but I already have a lot of experience. I wonder, if you're starting from scratch, whether that's a good idea. But I still think it's probably a really good idea, but basically generate some code using AI and figure out what it's doing by playing with different parts. Maybe can you comment on that aspect, like the use of AI as part of the learning process?

ThePrimeagen

This is where I have both the hopeful and the doomer take at the exact same time, and it's the same thing with Google or Stack Overflow, it's all the same kind of take, which is it's just making things more democratized in some sense. I get to ask questions in probably the most personal possible way with my own voice, in my own words, and it's able to produce out answers and hopefully help guide me. Now, regardless of just say the errors and the incorrectness of it, ultimately just using it as a learning tool and being able to just formulate and read answers in your own voice, I think is super powerful. And I think it's super amazing. But the part that I think is going to be really difficult is that we don't value remembering things anymore as a society. Since the internet came about, I can just look that up. I can just look that up. You don't need to memorize your times tables. You can just use a calculator. You can just do all that. I remember I just was sitting on the airplane and I watched someone do the world's most simple addition and subtraction like 10 times on their phone. And why are you not just... You should already know, you should be able to do these things. And I realized that we kind of offload our brains, right? Oh, I don't need to know these things because I can look them up. And that's not a bad answer in some sense. I can understand that. I don't need them to remember every last thing, but then it also makes me realize that you kind of develop this learned helplessness, that a new error comes up. I'll just ask the AI. AI says, oh, okay, I got to fix this line. I fix the line. You didn't actually learn anything. You kind of just used it as a quick means to get something out and move on. And so you sacrifice knowledge for speed, which is a great thing in some... We have to make those trade-offs all the time in engineering. Sometimes you have to move fast at the sacrifice of knowledge,

and I'm totally on board for that, but I worry that what we'll create is an entire generation of incompetent programmers who can do some amount of things well, but anything that is unique, bespoke, or requires some extra like little elbow grease, might become very difficult. It might cause a whole chasm where juniors remain juniors forever. And I don't want to see that. I want to see people grow. I want to see people actually be able to take this as a craftsmanship thing. And so that's both my hope and my worry is that AI think can do both really. If you could ask whatever question you want and you don't have to rely on, say, a book to give you that exact answer. And if the book just said it wrong and you can't understand it, it's just like, sorry, you don't get to learn what this is. Like recursion for me, I spent way too much time until someone gave me the right problem to understand recursion. You could imagine AI could have solved that for me way faster because it could have gave me the right problem and walked me through much better. But what happened if I just always have recursion solved by them and not actually learn it myself?

Lex Fridman

So if I ask AI to generate code to do a certain thing, actually a large percentage of time, most of what AI generates is going to be correct for me, but some percent of time it's not, fundamentally not. And for me to recognize the difference between those two, I think it takes a lot of experience. I think to learn that skill of knowing, no, no, no, a different new out of the box solution is needed here than the one you're providing. You're missing the point. That's a skill, and how do you learn that? You learn that by building from scratch. So both are probably really necessary. But I think as a first step of learning how to program, it's pretty nice to generate a function, to generate for loops and all that kind of stuff, and then just fuck with the different lines and modify them to try to adjust the behavior of the program, and from the way the behavior of the program adjusts or bugs are created, you learn about the syntax of the language, the behavior of the language, all that kind of stuff. So I think it's a super powerful way to learn. But yeah, you need to also write from scratch.

ThePrimeagen

At some point you have to take off the training wheels, because I think what you're really spotting is the difference between reading and writing code. I can read a lot of languages very well. I can see what's happening. I can understand it, but I would not be very good at writing it. I can understand a lot of things about C++ and I can read it, but I'm just not that because I just don't done it in so long. I can't remember where all the semicolons, and colons and you do public and private, and how should you do naming conventions? All those things kind of add all together, and then you're just like, oh, I'm really bad at writing it, though I can read it. And so there's a skill gap chasm that exists between those two.

Lex Fridman

All right. Well, let me talk about the various languages. The cheesy, ridiculous question of what's the best programming language? Let's say, what's the best programming language

that everybody should learn? Maybe let's go with the top five. I'm going to pull up the Stack Overflow developer survey, because I think we have... You don't like them?

ThePrimeagen

You got to remember, because I mean, you're a data guy. You know about biases and data. What does Stack Overflow naturally bias towards?

Lex Fridman

Well, they have the different slices of professional developers, junior developers, they have different slices. Okay, what is the bias?

ThePrimeagen

I hear you, but who fills out a Stack Overflow survey? Someone who participates on Stack Overflow. Who's participating on Stack Overflow? Largely very, very new people, and that one guy that loves answering questions. And so I'm not sure if Stack Overflow is a great place to get data. It could be a very biased set of data.

Lex Fridman

Is it really only new people?

ThePrimeagen

I mean that's who's using Stack Overflow.

Lex Fridman

All right. Most popular technologies. On this...

ThePrimeagen

JavaScript, HTML, Python, SQL.

Lex Fridman

SQL is the more general kind of... I'm sure they're not doing the individual sort of flavors of SQL. By the way, pronounce SQL versus Squeal?

ThePrimeagen

It's squeal.

Lex Fridman

Squeal? You squeal?

ThePrimeagen

Squeal, I think is the correct way.

Lex Fridman

Squeal.

ThePrimeagen

I did SQL because I didn't know the audience. I don't know if they can handle the truth, which is its squeal.

Lex Fridman

The squeal of joy, squeal...

ThePrimeagen

Squeal light, my squeal, Postgres squeal.

Lex Fridman

By the way, I had a lot of joy from earlier saying pigfucker, for some reason.

ThePrimeagen

It's such a [inaudible 02:31:36]. I mean, can you believe that was a real conversation that I had?

Lex Fridman

Yeah, that was. TypeScript, BAS, Java, C Sharp, C++, C-PHP.

ThePrimeagen

It largely kind of aligns with the world you'd expect, but Assembly, why is Assembly more popular than Ruby? Who's writing just Assembly by... No one writes Assembly by hand other than maybe that one guy that's developing TLS 1.3 and hand rolling a cryptography algorithm to be the fastest possible algorithm.

Lex Fridman

Yeah. Assembly is a weird one. Maybe people write it maybe in school, but even in school now for a operating systems course or something like that, or system engineering. I don't know if they write Assembly anymore. I don't think so anyway.

ThePrimeagen

And Swift and Ruby being less popular than Assembly seems ridiculous. But nonetheless, okay, so you get my ideas behind that, but as far as top five languages go, that's probably too broad because you could just name so many. I think you should probably archetype it by what do you want to do? So if you want to get into game development, perhaps C Sharp, C++ could be good choices. Or JavaScript and doing Canvas games, I could see that also working. But you're limited by doing JavaScript obviously, because you can't do as much because the language is just not fast enough to do as much. So it's like a good thing to

remember. If you're going to be doing backend stuff, if you want a job, if you're looking for a job, maybe C Sharp slash Java, or JavaScript, or Go would be great choices. If you're looking to do embedded, you probably want to do C, C++, like that would probably be a good choice. And so I think you have to first determine what do you really want to get out.

Lex Fridman

If you're just curious about programming, which I talked to a lot of people who are, yeah, you can consider jobs, but basically their question is, okay, what's the first language I should learn, and maybe what are the several languages I should explore?

ThePrimeagen

Can I say something that's going to make a lot of people angry?

Lex Fridman

Yeah, sure.

ThePrimeagen

I think the first language people should learn if they have no idea about anything is JavaScript.

Lex Fridman

Yeah. Why would that make people angry?

ThePrimeagen

Oh, because people just, first off, I'm not supposed to say anything nice about JavaScript.

Lex Fridman

Yeah, usually that's the meme, that you hate JavaScript, right?

ThePrimeagen

Yeah. No, JavaScript's a beautiful language, and it has a lot of things that are very great for it, and one of them is that you can express anything with very little effort. And so someone that's new, I think it's really great to be able to draw a box and move a box. That's great. You get to see it visually. I think that's one thing that's really great about JavaScript is that you can do that. Then you can go, okay, I want to learn about the backend. I want to make a request now. You can write a quick backend in it. Now you're starting to get familiar with programming a little bit. I can save this to a database. I can bring it down. I can put it on a screen and I can animate it all around, and I can even put it on a canvas and render it in 2D or 3D. So it's like there's so much variety of what you can do with JavaScript. It's a great way to get introduced into programming. But then at some point you have to go, okay, I now need to learn more about this whole thing.

Lex Fridman

I mean, yeah, just like you said, you can make games, you can do front end, backend for web development.

ThePrimeagen

You can even do embedded. They actually have... Like there's Wes Bos is building his Roomba or something and programming it with JavaScript and React, which is just the world's worst language to choose for embed, but you can still do it.

Lex Fridman

Also, we mentioned sort of in terms of applications, anything that relates to data or machine learning, Python is the sort of the leader there, so that's a great one.

ThePrimeagen

It seems like Python, CUDA stuff and C++ would be a dynamite in that, because a lot of these Python libraries are assumed you're just smuggling in C++ underneath the hood or C.

Lex Fridman

Okay, so JavaScript. I'll say Python.

ThePrimeagen

Python's a great one too. You can get quite far with it, but you can't write the front end. What happened if you love the front end? What happened if you really just want to design things and you just didn't know that?

Lex Fridman

Well, it's okay. So for that, JavaScript.

ThePrimeagen

But Python's a good choice because you can't do the ML stuff in JavaScript nearly as easy.

Lex Fridman

Do we count HTML and CSS as programming languages?

ThePrimeagen

I think there's some technical definition that it is. If you use this certain amalgamation of CSS plus HTML, it actually has, it can be a Turing complete language. But I mean for practical purposes, no. HTML is not a language. For me, yes, the Turing test is a good one, but for those that are just not wanting to be as academic, if I can't write a function and an if statement, I don't feel like that's... If I can't loop, if, and function, I don't feel like that's a good, that's a programming language.

Lex Fridman

Although modern HTML has a lot of features.

ThePrimeagen

It's crazy how much it has, but it's more of a specification than anything else. I specify it to be a pop-up. I specify it to have this kind of accessibility, this kind of look. Under these conditions look like this, transform like this, move down here.

Lex Fridman

I don't know. I kind of like these popular programming languages in this list. I like JavaScript.

ThePrimeagen

You like Bash?

Lex Fridman

Oh, yeah. I like Bash a lot. Yeah. Why?

ThePrimeagen

Okay. Bash is kind of one of those ones where it's like, do you really like it? I like it up until I need an array.

Lex Fridman

Oh, as a programming language, just no, but I like the command line.

ThePrimeagen

Okay.

Lex Fridman

Do you like Bash? No, nobody likes Bash. Do you mean-

ThePrimeagen

Someone is so offended right now.

Lex Fridman

It means do you use it a lot? Yes. I mean, it's good to learn, right? It is good to-

ThePrimeagen

It is.

Lex Fridman

... Be comfortable in the command line because it's a bit of a superpower. It's like, I think I follow on Twitter, FFMPEG.

ThePrimeagen

Great account.

Lex Fridman

There's certain Twitter accounts that are just legit. And I think FFMPEG, they have all these sort of parameters that you can add on the command line, that it's like one of those cryptic languages that only very few wizards understand. But once you begin to slowly understand, and I'm only at the very sort of beginning stage of that journey to mastery, the powers you gain at every step, it grows exponentially, it feels like. I mean, FFMPEG is just this incredible, what would you call a library system? There's just the people behind them must be just brilliant masterminds because they have to work with all these codecs, with all these containers, with all the mysteries of the media codec universe they're masters of. And they understand compression, which is another super fascinating technical set of problems that, I don't know, just FFMPEG just fills me with joy that it exists. But you need kind of Bash type comfort, command line comfort, to work with it to really unlock its power. Yeah.

ThePrimeagen

I think FFMPEG is probably one of the most consequential libraries of our day, and the Twitter account is so unhinged. It is the most amazing thing to see because I think FFMPEG does not get the love it deserves. Every single application, OBS, probably FFMPEG underneath the hood. Everything, FFMPEG underneath the hood, and yet they do not get the love they deserve. I just love it. I just think they're the best.

Lex Fridman

Yeah, I would say JavaScript, HTML, CSS, Python, SQL, I mean that is SQL Squeal is a programming language. It's an incredibly sophisticated programming language. Yeah?

ThePrimeagen

SQL is interesting. I believe you can classify it as a programming language. It does have, if. You have case statements and it's pretty crazy what you can do with it.

Lex Fridman

You could do functions, you can do all that stuff. You shouldn't.

ThePrimeagen

Yeah, for stored procedures, that's how you make your life hell. I will say that all the top languages there, none of them are strict static typed languages. And so even TypeScript, I don't like this any. And so for people that are learning, doing something that's much more

strict would be great. Something like Go, Rust, I mean even C Sharp, C++. Anything that kind of changes your perspective of types I think is really helpful to kind of go through. They're not getting nearly as much love on this most popular language list, but I think they're very fantastic.

Lex Fridman

All right, well, if I put a gun to your head, top five languages, let's list them out. There's a bright-eyed 20-year-old asking you, what are the top languages, five languages to learn?

ThePrimeagen

If I were to pick five languages that I think people should learn, or at least, let's restate it this way, I'm going to say a couple languages and you should at least explore some of them. I think you should explore a loosey language, so Python slash JavaScript, where there is truly only one type, which is a boxed value, which is a multivariate, different types underneath the hood.

Lex Fridman

What did you call it? A loosey language?

ThePrimeagen

A loosey-goosey language. It's a dynamic language, and so I think it's really good to explore one of those too. So I'd put Python or JavaScript right there. Even Lua, throw Lua in the bunch. I think you should explore a strict language, so I'd do something like Rust, Go. I think those are both really, really great.

Lex Fridman

C++?

ThePrimeagen

You can do C++. You can do some type erasure in C++. You can do it with Go as well, but for the most part it's a great language to do that in. It can get a little wild. New C++ seems great. Everyone keeps telling me new C++ is great. It has every feature you've ever wanted and all the features you don't want.

Lex Fridman

Yeah, exactly. I mean there's smart pointers, there's dumb pointers, there's all kinds of pointers. There's no memory leaks. That's not an issue.

ThePrimeagen

Foot guns, face guns, soft beds. There's everything in there.

Lex Fridman

Unless you like memory leaks, it has that too if you want that kind of thing. It's great.

ThePrimeagen

Okay. How about this one? Languages that I actually want to really learn, that at least sit in my curiosity bank. There's three languages, which is going to be Swift, Elixir, OCaml, and then I'm going to throw Odin in there, just because Ginger Bill is great. But Elixir and OCaml, I don't have a strong functional language underneath my belt. That's something that I just genuinely lack.

Lex Fridman

Yeah, I've heard incredible things about Elixir, about Odin, about OCaml. Obviously, I'm a person, as you know, who loves Lisp.

ThePrimeagen

I have never done Lisp. Lisp could be in that category too, just, or Closure I think at this point is what everyone tells you to use.

Lex Fridman

So in the case of Lisp, I don't want to speak negatively about Lisp, but it's important about modern community, what the community looks like. It seems like there's an excited, maybe small, but an excited community around Elixir, Odin, and OCaml, so that helps. Because then you can post shit on Twitter that you're like, I accomplished this. People get excited and it's nice. It's a good feeling.

ThePrimeagen

You can post something on Twitter and you'll get a thousand likes if you do something cool on Elixir, which is that's a pretty big amount of people to like a post for such a niche topic. Programming's already a pretty small topic. Then you get into functional program. That's a small topic in a small topic.

Lex Fridman

Yeah. I don't get that much. If I post something about Emacs, I'll get crickets if I post something about... If I proudly use Neovim, there'd be a lot of people like, good job.

ThePrimeagen

Because it is the best editor.

Lex Fridman

Yeah, maybe it's just hype.

ThePrimeagen

Come back to the Civil War, Lex.

Lex Fridman

Yeah. Sometimes you have to sacrifice and go from the superior editor that is Emacs and choose Neovim just to be popular. You sacrifice integrity and values and quality for just popularity. It's a choice you made.

ThePrimeagen

Absolutely. I love how you put it.

Lex Fridman

Okay. Anyway, what were we talking about? I like how you're doing this in bunches. That's great.

ThePrimeagen

Right now, my kind of side honeys that I'm exploring-

Lex Fridman

Side honey?

ThePrimeagen

Yeah, side honeys. They're not my mainstay. Right now Go's kind of my favorite one to build a web app in. If I'm going to build some sort of backend with a lot of complicated logic, Go's just so convenient. But I get really frustrated with its ability to express everything that I need. If you have a list, a heterogeneous list, a list that contains two types, Go's just really not that fun to use. And I could see, so the ones I'm exploring is Jai or J, or the language as Jonathan Blow says, and Zig. And both of them have a lot of power to them. They're both very interesting. They definitely have foot guns in them. They're definitely more, they don't take it easy on you. Zig seems like it's a really amazing language, and so does Jai. They're both very cool.

Lex Fridman

Yeah. Actually, I saw Dave Plummer's testing of close to 100 languages for speed, and Zig came out on top.

ThePrimeagen

That was a mistake. I mean, when I say mistake, nothing against Dave Plummer. He's an extremely talented engineer. It's just that Zig, C, C++, all those languages that were being tested, they're all LLVM backends, right? That's the one that actually turns the thing into the executable part. And if there's a variation in speed, it just means in one language you didn't quite express what you are supposed to correctly. There's the language ball test that's been

bouncing around on Twitter. Zig was like sixth or seventh below I forget what language it is. I played around with the example, added the word “no alias” to the argument, which means that the piece of memory that’s coming into this function, there’s no global pointers, there’s nothing to it, and so the compiler can make these really cool optimizations. And I made it faster than the C version. So it just means that it’s just not correctly specified is all that means.

Lex Fridman

Yeah, but it’s still exciting. To me, the competition between Zig, Rust, and C++ is really interesting. Part of it’s for speed. Part of it’s how easy it is to write performant code.

ThePrimeagen

I’ll say something. That’s the reason why I think Zig is so interesting comparatively to say C or Rust. C is the ultimate language. It can do anything, have pre-process or macros. You can do quite a bit with it, but it’s also really difficult. So macros, you can do quite a bit with it, but it’s also really difficult and it’s also really simple and you can learn it. So it’s kind of its own unique beast. And when you get really good at C, C is a magical language and people are really great at it and people speak very highly of it. Rust is like this ultra safe language. What you can do in C, you just can’t even express in Rust. Rust is going to be the safe man that holds you at night, keeping you warm, right? It’s going to be just the greatest. But somewhere in the middle lies Zig. Zig has optionals. If you’re not familiar with optionals, that just simply means there’s a value here or there’s not, but you first have to check that before you can use it. So it prevents that whole null pointer dereferencing segfault problem. And that’s not available in C, just by default, you have to kind of build that thing in. It is the only option in Rust, but Zig says, “Hey, if you have a pointer, you can’t express it as null unless if you market that it can be null.” There’s ways around it, there’s other types of pointers and stuff like that that can do that. But for the most part, Zig will give you safety for the most part. So it’s like a little bit of safety, but more like C. So it kind of gives you everything you want in that region where you can express safe code and unsafe code. It’s very easy to write. It’s very pretty. Or at least the idea behind it is very pretty. The language itself is bland, but.

Lex Fridman

Wow, there’s beauty in everything, Prime.

ThePrimeagen

Yeah.

Lex Fridman

You’ve programmed in Rust a lot. What do you love about Rust? What are the strengths? What are the weaknesses? Maybe you can speak about memory management that you

already mentioned, the challenge of memory management that several of these languages address, but yeah, what do you love about Rust?

ThePrimeagen

What I love about Rust, I love the ability to free the memory that you're using is directly tied to the stack. So whenever you create something, there's a stack variable or there's some amount of stack memory, whether it's a pointer off to the heap, a pointer and a length. So some amount of memory on the stack and then some memory on the heap because a string is not all on the stack, it's some on the heap, some on the stack. And when that stack variable goes out of scope and gets cleaned up, it also cleans up what's on the heap. So it kind of simplifies this whole idea of, whoops, I forgot to free my memory. It just does it for you. So it's not a garbage collector, which will do it sometime later. It's not like C where you have to call it yourself, it's somewhere in between. Now, there's a lot of strategies people use, arenas and all that that make that C part much easier. I'm just not even mentioning it, but it just makes it a lot easier. But Rust does that really beautifully and it's just like a really cool idea about it and I really like that. And the second thing that I think Rust does really, is such a good thing is that mutability of something is you have to specify it. So you don't just create a variable and then mutate it. You have to say this is not only a variable, it's a mutable variable. And I think that just makes code really readable and really understandable. Because anything that does not have the word `mut` next to it, you know for a fact it cannot change. There's some rules around that, but you get the general idea.

Lex Fridman

Unlike most programming languages, you have to explicitly state that this is going to be changed.

ThePrimeagen

Yeah.

Lex Fridman

Yeah. That's really interesting. I mean it's safe, it's trying to be, and the safety might be, create limitations. Let us consult the AI overlords. Rust is a blazing fast memory efficient systems programming language that emphasizes performance, type safety and concurrency. The language enforces memory safety without using a garbage collector, as you said, instead utilizing the unique, quote, "borrow checker" that tracks object lifetimes at compile time. This prevents common programming errors like null point or dereferencing and memory leaks and so on. Yeah. So you've also spoken about metaprogramming. Which of these languages do you like for the metaprogramming? I love metaprogramming in C++, but it's a giant mess. At least when I program C++ 17 standard, I believe, it's just a mess, especially a mess to debug.

ThePrimeagen

Yeah, I would consider myself kind of a metaprogramming newbie. I have only solved some amount of problems with it. That's kind of like what this year is for, is for me to really, I want to see where the ends can go in that. So I don't have a strong opinion on this one. Zig, one thing I really like about Zig is that the metaprogramming is also the language itself. So you don't have to, there's not an alternative. So with Rust there's an alternative. When you create a macro, you have to do the macro syntax. With Zig, it's just, it is the thing, you just program it and you add the word comp time if you want it to be a compile time only. So you can create the list of prime numbers at compile time in Zig, which is kind of an interesting, unique thing. So you have code that executes at compile time and then you can take advantage of the result of it at runtime. So neat, right? That's how I'd look at it. But again, I haven't used it to the point where I feel like I can super authoritatively talk about it.

Lex Fridman

You have been undecided, what language are you going for this year?

ThePrimeagen

I'm going to keep Go as my mainstay, my two side honeys, Jai and Zig. I'm going to explore and try to build out a service in them that can do a bunch of talking to, say, ChatGPT and ElevenLabs and send stuff down to client and work with web sockets. And I want to make sure that, I just want to see how do they perform in this realm. And I may be using the language incorrectly, like Jai, it's not really been designed for the web world. I just got done writing the ability to read Twitch Chat and it required me to do Berkeley sockets. So if you're unfamiliar with Berkeley sockets, it's like the old way of doing it, it's how you do it in C. So you have to kind of go through the whole nine yards of creating your own connection. I had to create my own connection, I have to read from the socket, then I have to parse out all the IRC, right? You have to kind of build it from scratch. There's not like a new TCP connection to this server. You have to be like, "I'm creating a socket." You're going to be of the IPv4 family and TCP and you're going to do, you know, I'm going to now have to take your address and go look up your address with DNS, get that address back and then connect to a TCP. So it's a lot more manual still. It's a lot more raw in that area, but it's fun.

Lex Fridman

What are some epic projects you've built on stream that jump to memory?

ThePrimeagen

My most favorite, sorry for interrupting you. So I'm really jazzed right now.

Lex Fridman

Let's go.

ThePrimeagen

Okay. So jazzed.

Lex Fridman

Jazz hands.

ThePrimeagen

My most favorite project was the one I did last year. Someone built a Doom ASCII port. So you could play Doom with ASCII. So that means you could play it in your terminal. Very, very fun, very excite. So I made a Go program that could spawn out the Doom ASCII, then I took that Doom ASCII and I sent it to the browser so that people could play Doom ASCII in the browser, but then I made it so that Twitch chat could control that instance of Doom ASCII by piping in Twitch chat, taking the average of the movements over so much time and replaying it as if it was a controller. And I had Twitch chat beat level one by spamming it. But the fun part was I used a bunch of fun encoding techniques. I used quad trees to be able to take smaller amounts to use run length in coding. Tried to create my own compression algorithm because if you're sending out a bunch of ASCII stuff, it's still pretty expensive because you have to represent color, color's not cheap. On top of it you have to represent what does it look like? What does the ASCII look like? Well, I realized there's all these fun techniques you can do for compression like the shape of the ASCII you send down in a lot of these engines are actually just proportional to the lumosity of that pixel. So you'd use an eight to represent or a pound sign to represent white, but black, you're going to want to do a period or a comma or a bar, something smaller. So it's like I then developed all these different compression algorithms that turn a bunch of data, which would take, I forget how much it would take. It'd take gigabytes upon gigabytes to be able to send out to thousands of people to all see the same image at the same time, to all be able to interact with Doom at the same time. I turned it from gigabytes into kilobytes by just trying to figure out how to make it as small as possible and send it all out. It was super fun. Absolutely had a great time.

Lex Fridman

So you're actually sending it to all the people in chat. So where's that pipeline, how chat is able to control the Doom thing?

ThePrimeagen

Twitch chat. Yeah, so they would go, people would spam W and if you said W, it would hold down W for 150 milliseconds if the majority of people during that time period said W.

Lex Fridman

Nice. Okay. And how are they getting the input of where you are on screen?

ThePrimeagen

So originally I was going to send that through Twitch, but Twitch is like five seconds behind, so that's why I piped it out to a website so everybody could see from my computer to the website and typical lag was right around 70 milliseconds. So it's like they could mostly see what was happening in that short period of time. It was pretty exciting. So we had 1,000 people, or I had somewhere between 1,000 to 1,400 people smashing Ws and pressing F to fire and turning and we killed some zombies. We blew up the barrel at the very end of level one to kill the imp.

Lex Fridman

How are you getting the Ws from the Twitch chat? Is there an API?

ThePrimeagen

I was using IRC, so just a little TCP socket and then you just parse out IRC.

Lex Fridman

Okay. And there's very little lag there. Okay.

ThePrimeagen

Yeah, I think it's a couple hundred milliseconds though. It's enough that it actually made it a little bit difficult because people would often overturn and then go forward and miss the door and then they had to go back and...

Lex Fridman

That's awesome.

ThePrimeagen

It was awesome. So that was my favorite I think project of all time just because I never got to do a lot of encoding. Encoding's kind of like, what do you normally do? Okay, I need to send something down. I don't know, gzip it, server will just do it. Server just does the right thing. I don't need to think about it. So instead it's like I think about it, I'm going to send the right thing.

Lex Fridman

Yeah, you have to think about the compression. Yeah. And there you go. That's some more love towards FFmpeg because they have to think about that a lot.

ThePrimeagen

Ultimately inspired by FFmpeg and their awesomeness.

Lex Fridman

So can you speak to just the chat community in general? A big part of what you do in terms of streaming is the humans that are communicating with you live. Can you talk to the different chat communities? First of all, which is the best chat community, YouTube, Twitch, or X?

ThePrimeagen

This is where I feel bad for YouTube, because I do think it's technically the worst, but it's not YouTube's fault. And let me kind of explain why.

Lex Fridman

And then I will explain why you're wrong. But go ahead. YouTube is great.

ThePrimeagen

I know you love YouTube but let explain why, is that when you go on Twitch, you go to anyone's channel, they have this cultural human centipede thing that's happening where as the memes flow in, all of Twitch reacts and morphs to all those memes. So every channel you go to has this same culture. There's a lot of similar emotes and everything, so it's very tight-knit. So when I stream, I get all the same jokes that you would pretty much see if you saw, I don't know, Sodapoppin or some big streamer, Asmongold, whoever, [inaudible 02:55:56] software streaming. All the same memes would all flow through the exact same kind of pipe. And so it's a very holistic kind of community. So every time you're making jokes, you're making jokes that are in the ether. Twitter kind of has that too. Tech Twitter kind of has a set of jokes and so you can kind of see it. The problem with Twitter chat is that there's just nobody there right now. Typically just to put it into perspective, I have somewhere between... somewhere between like 1,500 to 3,000 people on Twitch, somewhere between 800 to 2,000 on YouTube, and like 50 people on Twitter. So the difference is massive. But Twitter has that same thing that's developing where there's memes that are constantly flowing through it. And so they're very highly connected. YouTube just doesn't seem to have that. They're just a bunch of people and people go to YouTube for various reasons. I'm going to YouTube to learn. So they come in and they want to learn. So they're not on the meme train, they're not in this cultural zeitgeist train. They're just like, "But why would you use this if statement when a switch statement in this one particular case?" And you're just like, well that's not what I'm trying to do here.

Lex Fridman

Yeah, you want to captain the meme train or you want to ride on the meme train.

ThePrimeagen

Yeah. Or you just want to be able to create a culture on your chat because your chat's going to be some variation of that kind of zeitgeist that's flowing through Twitch. And it kind of is very contiguous between X and Twitch. It just feels really out of sync with YouTube. And

then YouTube particularly does a bad job. And some people would argue a good job because you can swim. Swim being, you can actually change what timestamp you're at. So all of a sudden you'll be like, oh yeah, something about driving to soccer in my minivan. And then 20 minutes later you'll be talking about Zig and then someone's like, "I personally use whatever to drive to soccer." And you're like, "What are we talking about?" So YouTube is a very disjointed chat as well because it depends on where they're at within the video. Swim comes from Netflix, by the way, call it swim.

Lex Fridman

The term?

ThePrimeagen

Yeah, people said swim.

Lex Fridman

Oh, so you're, okay.

ThePrimeagen

Swimming through-

Lex Fridman

Yeah. So you're not just making up the term. Thank you. Wow.

ThePrimeagen

Yeah, but it's probably made up and probably only 10 people said it at Netflix, so no one's going to know it and they're going to be like, "Yeah, right. That doesn't happen on Netflix."

Lex Fridman

So going back to projects, what projects on stream or in general?

ThePrimeagen

No, you need to answer why YouTube chat's the best chat.

Lex Fridman

Well you kind of convinced me. Okay, why YouTube is the best chat. Well, I think I'm just a hater. That's basically what it boils down to and I'm just talking shit.

ThePrimeagen

Love it.

Lex Fridman

And I'm probably just from the outside shooting in because Twitch is such a fun culture of memes. And so it's just fun to shoot from the outside to egg the house of Twitch. And then I just sit back on my lawn chair with the small YouTube community just talking shit. No, you're absolutely right. There's a real sense of community that Twitch can form. But I just like the openness of YouTube. It's just better at opening to the world. It's more accessible, it's easier to share. It's just a more established platform, that's all.

ThePrimeagen

Fully on that team.

Lex Fridman

For the open world. I can send it to people that don't usually watch video game streaming or that kind of stuff.

ThePrimeagen

Yeah. If you send a Twitch link, they're like, "I don't like video games." And you're like, "Well actually it's not video games." That talk happens every single time you mention Twitch because Twitch does have a perspective about it that YouTube does not.

Lex Fridman

I was just on Joe Rogan's podcast and I think it came up, he asked something like, "Is Twitch still a thing?" So that just gives you an example. And then Jamie said, "Yeah, yeah, it's definitely still a thing. It's still growing and so on." And so yeah, there's just a big slice of humans that don't participate in the Twitch Twitch sphere. Yeah, I just like talking shit so yeah.

ThePrimeagen

That's a beautiful answer.

Lex Fridman

But it's cool that you sort of make it accessible on all these different platforms. And I have high hopes for X, but yeah, it's feature-wise still has a lot of growing up to do.

ThePrimeagen

And just why do people use X? You typically are going there for a text-based interaction you want to look through. So I also think they just have a user expectation change that needs to happen. And that just takes a while. That's going to take a little bit before people get to it. I think their idea of audio first is a great first step where people can listen to it and have the phone away maybe. There's a lot of changes that have to happen before X can be successful in that.

Lex Fridman

I mean, X has this incredible comment section just like Reddit, right? So it's like-

ThePrimeagen

You said incredible. That's not Reddit. Comment section, correct.

Lex Fridman

Comment, yeah. Incredibly dynamic and vibrant even if it's... Yeah. What is the technological platform? How does the interface and the technology shape the discourse? It's fascinating because X has a different style than Reddit, different style than Facebook, different style than Instagram. It's interesting. And all those common sections are different technologically, like how the sorting is done, how easy it is to sort of build a community around it? Because YouTube is not really a community. Every single video on YouTube has its own mini community. You're all talking on just that one video. But you can't jump across.

ThePrimeagen

There's not like, "Hey Bill, hey George." There's no crosstalk that happens in multiple videos.

Lex Fridman

Yeah. But the community is awesome. I love community. I love the feeling of community and I guess that's what Twitch really provides.

ThePrimeagen

YouTube also does have it though. They have an aggregate community. There's a lot of fun comments and all that on the videos and a lot of thumbs up and then you see the fun discourse that happens and it's like that's the community, it's just only a certain slice sees it.

Lex Fridman

I think that's even more so on YouTube for live-streaming. All the same folks show up and they talk shit, they celebrate, the meme train arrives.

ThePrimeagen

Yeah.

Lex Fridman

Okay. So now, what projects shape you as a programmer? Whether the ones you streamed or offline.

ThePrimeagen

For me, I don't know if there's a one project I can point to, but I can point to a specific spot where I think it happens and where I think you can learn a lot from. Any small program you write will be somewhere between 1,000 to 5,000 lines of code I consider a pretty dang small

project. You can correlate this to any feature within a larger system as well. A specific feature on a website could be a thousand lines, a couple thousand lines. There's a point in which all of your choices add up. And I typically find that right around 5 to 10,000 lines of code. The choices you've made either weigh you down or kind of free you up. And so it's right in that, that I feel like I learned the most is because I love getting to that point in a project or in some small part of the code base because at that point I get to test, A, how good were my initial gut decisions about how I designed the software, but B, now I need to go back and think about how am I going to do testing across this in a more effective way? How can I scale this out to 20,000 lines of code? How can I do all these things with what I've got or do I need to kind of rethink it? And I find that that's really where the best learning happens is that everybody has probably a different number that exists, and as you go to each one of these numbers or how well or holistic you want your project to be, I think that you'll come up with different numbers. And I think that number should just get bigger as you get more experienced. Because there's projects that are a million lines of code, but they're most certainly not holistic, right? Every part of the code base is some age at some capsule of time with some sort of programming style. Some is more functional, more class-based, more, God help your soul if it's pre-processor macros in C++. There's all these different kind of things you'll find throughout time. And so that's why I try to think about it as the feature or the thing you're working on. It's usually about 5,000 lines is where I find that things get kind of, did I make good or bad decisions? And that's where I do all my learning is right on that phase. I'm trying to get it to the point where I should be able to shoot from the hip and do 20,000 lines and not be upset about it.

Lex Fridman

So first of all, just enjoying the thing you create part, yeah. About there you can sit back and see all the parts dancing together. For me, also debugging, you get to see the choices you make materialize as how easy it is to debug. I'm a big proponent, I think you've mentioned this in the past, I put asserts everywhere.

ThePrimeagen

No, you are the reason why I do that.

Lex Fridman

Yeah.

ThePrimeagen

You were like the first one. Keep on going, sorry.

Lex Fridman

Really? Okay. So for me, one of the joys, whether it's try catch box, whether it's assert, whether it's with the testing, I get to see the payoff of all the minefield of asserts I've laid out before me in my kingdom by how quickly I can debug a system as it grows larger. And I

can first of all discover errors before they become real bugs and also how quickly I can solve those errors. And that brings me joy. For me, a lot of the joys of programming is creating powerful systems that don't break down, that work correctly, that work correctly in majority of the cases. And there, sort of the stress testing the system and getting all of the signals from that system that everything is working correctly is something that fills me with joy and makes sure that the system actually works. So yeah, that, I don't know if it's 5, 10,000 lines of code, if it's Java or C++ it's millions lines of code. But yeah, in Python, yeah, I would say 10,000 lines of code. That's when you first get to see the magic. But anyway, you were saying?

ThePrimeagen

Okay, so you and John Carmack had a conversation about asserts.

Lex Fridman

Yes.

ThePrimeagen

You talked about this idea of putting asserts everywhere that effectively crash the program when you have some state in your program that should not be represented and you have made this choice actively. And so I've never done that before. And I know this is like an old technique and I obviously must be too young or too dumb to know that this was a thing people did. I grew up in Java and I think that's probably why I didn't run into this. So I saw that and I was like, I'm curious about how to use asserts more. And then I ran into a person named Joran. He's the CEO and creator of TigerBeetle. It's like the world's fastest, greatest financial database. And it was spawned out of a company that needed to do a bunch of financial transactions. And it's written in Zig and what they do is they do deterministic simulation testing and they just use NASA's kind of guarantee for creating really great software. So don't use U size, specify your exact size of int you expect everywhere. All these kind of things they do to be very specific. And one of them is that every function should contain two asserts. Whether it's positive space like these things should happen or negative space, like this pointer should never be null. You're programming into things that should never happen. Normally, you would just never specify that. You'd never think about that. So every single function everywhere has all these asserts and these asserts run both in production and in testing. They're always on. And then they take deterministic simulation testing and run like 200 years of just random data, just complete slop going through the system and seeing how far it goes. And when an assert happens, they're like, here's the input that caused it, here's every last little bit that happened, and now you can identify where this went wrong. And it was so cool. So between you, John Carmack and Joran, that's where I got like, okay, I got to really... And NASA, I'll throw NASA a bone as well. NASA can join in on that one. I was like, okay, I want to try this. And I did try it. I built this big reverse proxy for me trying to do some game development stuff. And I just went ham on the asserts. And then I built the whole simulation testing thing that could do everything

deterministically. So even the result of requests would all come in specific orders. And I found a bunch of bugs that I just would never have found. And then I did it for a game I was making. I found some bugs where my cursor went off-screen, it would cause all these different problems because I just never tested them. And it's super fun and it's like a really great way to program.

Lex Fridman

Yeah, I think it's a skillset you grow over time. It's not just that you have to specify the preconditions, everything that has to be true, it's also adding things that are like, you might not even think about. You have to sort of anticipate really weird things. And if you add asserts, especially in complicated functions or in complicated classes that are able to catch really weird things, that's going to save you so many headaches and it's going to help you learn about your own code. This is one of the things, I think it was Jonathan Blow that either in conversation with you or was it in a presentation, he said that when he's starting in a project, he usually doesn't know how to implement it, how it's going to work. And I think he was saying that he wants a programming language. This might have been a criticism of C++, I'm not sure, where he wants a programming language that makes it as painless as possible for him to not know what he's doing, how he's going to implement it, and to quickly get to a place where he figures it out. I think there's a fundamental part of programming is building stuff while not really knowing what the next thing you're doing is. You kind of have a loose design, maybe a strict design, but really you're solving puzzles that are not... It is a dark room in a fundamental sense. And there you have to anticipate the kind of weirdnesses that might emerge while not really knowing everything. Just this full fog, fog of war. And there that's a real skill to anticipate the kind of issues that might arise and put asserts on top of them. And it's also like spiritually, for me, been a really nice way of programming a building of living life as having very strict asserts that say, "You're going to fix this problem if it ever arises. You can't just look the other way." This idea of treating warnings as errors. Make sure your code compiles without any warnings. That was a big leap for me. It's like, but there's so many of them and it's not really that important. It's like, no, no, no warnings. Make sure you treat every single problem, even fuzzy problems seriously, because that's actually long-term is going to create code that's much easier to work with, much more fun to work with, much more robust, resilient to all kinds of weirdnesses, all that kind of stuff. So it's a different way of approaching coding, probably more NASA-like versus web programming style. But yeah, it has made programming for me personally, much more fun because one of the most painful things about programming is creating when you get past 10,000, 20,000 lines of code and you have to find a bug. And that bug can take hours, it could take days to find, and that's torture.

ThePrimeagen

Yeah. When your system gets sufficiently large, some of these bugs are just, they are very difficult. Bless anyone's soul that's working on million line code bases, because it does. I can't tell you how many times I've spent multiple days just trying to figure out the root cause

of the bug. Not even the fix. Just like why does this happen? And that's hard. So I love that. I just love the asserts because I'm not good at them, I can see it's definitely a skill that I don't put into practice constantly, which means it's just not like a muscle memory type thing. And so it's just one of those things I just love. It's such a fascinating way to approach a problem, because I would've never thought, you know what I'm going to do? If I'm wrong, I'm going to crash this thing and I'm going to crash it right here because I should never be wrong. But instead you're like, "Oh, actually that makes perfect sense. I should crash this thing. I've done something terribly wrong here. Why would this ever exist?" And then you're like, "This is going to solve a whole class of problems."

Lex Fridman

Yeah. And especially if it's in production, it's like, well a user's going to see this crash. It's like, yeah, well you should minimize the number of times any user ever sees the crash, not by having a nice blue screen or whatever the fuck, but actually stopping everything. And that's going to create an incentive for you to never have that happen. You're actually going to put in the time to make sure it never happens.

ThePrimeagen

And the nice part is with the web and all that, you can always pop up something and say, "Hey, things have gone very, very wrong or unable to recover." You can give them a nice message and then log it off so you can see it, and then measure how often are you doing it. I understand that there's a bit of interestingness to a web project like do you want to always crash a server? There's a bit of a gamble if you release a bad version and you crash all your servers constantly. That's a pain you're going to have to accept.

Lex Fridman

I think this is more applicable for single systems like robots and so on. You have struggled with ADHD. I think a lot of people are really inspired by the fact that you're able to be productive and flourish while having ADHD. How'd you overcome it?

ThePrimeagen

Well, there's a lot of things that ADHD affects and so I'll start with some of the easiest things, because there's directly applicable, then these kind of collateral damage applicable things that happen. So one thing that has really helped me with ADHD is maturity. I think that's just a thing that everyone needs more of. Meaning that I found myself getting so wiggly and so out of control when I would try to sit down and read, and I just couldn't handle it. I just felt like I'd read a page and didn't read anything. The part of me that just went, "Oh man, gosh, I just can't even do this." I had to just simply quit listening to it and said, "Nope, I'm rereading this page." I remember reading some pages in college like 18 times in a row, just like I'm going to force myself to just do this the correct way. And so there's an aspect of maturity that really helps, no matter what, I will do the thing I'm going to do and I'm going to do it well and maybe it takes me a lot longer and that's okay. That's not the point of it. It's

that I'm doing it and that's the point. And so that's one thing that I think just generally helps. And ADHD, no ADHD, the resilience, emotional resilience is just a really important aspect that just helps. And so I think that has been a large part that really helps me. There's things that I still obviously struggle with. It's clear where I'm really bad at stuff, and just trying to think through all the different things that I'm bad at. There's more things I'm bad at than I'm good at. And so programming obviously has something that just allows me to remain focused and it's like a strength of mine. And so I started off where I could just do it for a little bit and then just through kind of that emotional resilience, I was able to start doing it more and more. And so now I can just do it for like 10, 12, 15 hours at a time and I absolutely love it. And so it's become kind of like a joy. It's like playing a musical instrument. I'm really into it. But then if it came down to, "Hey, you need to go schedule your own dentistry and go do all these other things or make sure the kids have this type of stuff ready for the meals you need to pack throughout the week." I'm historically very bad at that and will probably continue to be very bad at that. And so I must say that one of the reasons why I excel so much is because I also have a wife who is so good to me and she helps clear out a lot of the things in my life that cause a lot of me kind of getting snowballed into a weird spot where I'm just distracted getting nothing done. And so she's really helped me. So it would be foolish of me to claim that I've defeated the ADHD by myself, but instead I find that the places that I can really control I've done a very good job at, and the things that I obviously need to do much better at, my wife has helped me a whole bunch. And so I've kind of cheated. Maybe I found a cheat code, a loving wife. But that has been the thing that has really helped.

Lex Fridman

You said a lot of interesting things. So on the reading and for me it's also audiobook side, I do the same thing and I've gotten much better at it, which is I tune out mentally and I read a page and you don't understand anything on the page. You didn't actually read it. And yeah, I forced myself to just reread it or re-listen to an audiobook, which is a much more common problem for me now, and forcing myself to really pay attention. Because I listen to audiobooks often when I run and it's so easy to just tune out. It's a skill. I didn't realize how much of a skill listening to an audiobook is, especially when there's other sensory inputs when you run. So I have to force myself to really pay attention to every single word. And if I don't, like tune out and don't remember what I just listened to in the past 30 seconds, I force myself to re-listen to it. And sometimes that means five times until, it's like punishing myself to like, "You're going to listen to this boring shit over and over until you get good at that little skill of like zoom in." And you're like, yeah, there's people, they're doing stuff, there's nature, it doesn't matter... Yeah, there's people, they're like doing stuff, there's nature, it doesn't matter. You're listening to every single word and loading it in and trying to stay focused, even there's just so many distractions all around you. Yeah.

ThePrimeagen

It's definitely a learned skill and it takes a lot of time. And when I say, "Oh, I was able to do from here to here," I'm speaking over the course of like five years of doing this every day. It's not some small... There's no... The nice part about that decision though is you can make that decision today. You can make it right now. You're going to be like, "From here on out, I'll never make that mistake again. I will say I'm going to read 50 pages, I will sit down and read 50 pages, and when I get distracted I'll go back to the last place I remember and I'll start again." And like that's a decision you can make. That's a mature, non-emotional decision to make. And you can do that, it just may be really painful for the first couple years of making said decisions. And then it gets easier and then it gets easier, and then it becomes more natural to change yourself.

Lex Fridman

Yeah. And with every medium, with every platform, I think it's like a new skill. For me, like using social media has been that, just like I end up like doom scrolling too easily on platforms. And one solution is not to look at all, which is kind of what I lean on mostly these days, but I feel like I should be able to check, just read, okay, feel a thing, learn a thing, and then put it down, versus you have this glazed look over your eye and you're not really paying attention anymore and you're dead inside and you feel horrible afterwards. I don't understand.

ThePrimeagen

The horrible afterwards is real serious. I've definitely... I can 100% notice that I am a more anxious person the more time I spend scrolling.

Lex Fridman

Yeah, yeah.

ThePrimeagen

I can just feel it. It's like something inside of me that's kind of... I don't know how to say it other than it like wants to get out but I don't really know what that is. It's not anger, but it's not... It's very anxious.

Lex Fridman

It's like the opposite of the feeling I have when I wake up in the morning and I'm feeling good, and I look out in nature and look at the sun and just, and there's like a bird chirping and this kind of thing. Scrolling through social media, even if it's like super positive stuff or whatever, it's still not the same feeling as the bird chirping. Bird chirping on Instagram is a different bird chirping than in real life, cause bird chirping on Instagram, I'll start swiping until there's demons of different types fighting inside my head and then different anxiety, insecurity, whatever the hell. Just the mixture of chaos versus the bird chirping in real life. That is beautiful. But again, that's the same thing as with the audiobook. It boils down to...

Man, these people that talk about meditation, I think that's probably... they're onto something, because that's what it is be able to like focus calmly and deliberately on a thing, whether it's reading or audiobook or existence. When they sort of observe the breath, you're able to silent out everything else and remove everything else from focus. Yeah. That's a skill. That's a skill.

ThePrimeagen

I heard it put really beautifully, which is that we in America really have misunderstood liberty because we typically have liberty as just the freedom to do whatever you want. And the argument was that it's not the freedom to do whatever you want, it's the freedom to be able to do what you will. And how often is what you actually want to do, you don't do because you get trapped doing something that you've convinced yourself in this quick moment you want to do? And so it's like, "I want liberty. I want the ability to control my energy and to be able to do the thing I want to do, not to get distracted and destroyed in all the millions of distractions." And some of us get handed a worse deck of cards, some of us get a better deck of cards, but I don't think there's anybody that doesn't struggle with it in the technological age.

Lex Fridman

Yeah, and that's the skill. What can you say to the skill of achieving focus in programming? Do you have a process of how you sit down and try to sort of approach a problem? So, all the different, not just distractions but the challenges of starting a project, of thinking through the design, how to maintain real focus, because it's really difficult intellectual endeavor.

ThePrimeagen

At this point I'm lucky, but when I first started I can remember that every last part of programming. I had to go look up, I had to go read, I had side quests at all time. Every step was a side quest. Why is my screen blinking when I'm trying to render this thing out? Oh, I didn't know about double buffering. Why is this happening? How do I even write to the screen? How do I... Everything was a question. I had more questions than answers. And so I constantly had the problem of side quests, and I find that to be a very exhausting thing. But as I learned my instrument very, very well, I don't have as many side quests. I become more and more able to just focus on the thing I want to do. And I find that to be something that is just super, super useful. So, when I say I'm kind of lucky, meaning that I've spent so much of my life preparing for this moment that now when I have the opportunity to do something, I can just do that thing and I don't... Like I can be just on an airplane and I can just program for hours. I don't have to look up a single thing. I don't have to do anything. I don't even have to test the code. I can write 1,000 lines of code on an airplane and I'm very confident that it's going to be 98% pretty dang good. And I'm very happy about that because that allows me just to be in the moment solving the problem I'm trying to solve. Then I have 100% of my brain power solving a problem. And this is why I also... It's the same reason why I recommend learning how to type and learning your editor so well you don't even have to

think about the action because the people that have to... Even if you just look down, that's still mental processing power you have to spend looking at a keyboard in which you already know where the key is. You do. At this point, if you've been typing for thousands of hours, you know where the key is, just stop looking down, you'll learn really quickly. And so it's like this thing where it's like, "I'm not going to spend all that time and all that mental effort looking up the thing. I'm going to just memorize... I'm just going to get it in me, and then I can go fast." And it feels good. And so that's how I'd kind of defeat that is because now I get to do something where it's like there's no more questions. It's now me just expressing myself into this medium and it feels really good.

Lex Fridman

I'm sure there's still like things that pull at you, like curiosities, distractions, like, "Ooh, I wonder how..." Anytime you have access to the internet, you're going to get-

ThePrimeagen

Twitter's a big one on that one. Yeah.

Lex Fridman

... you're going to get curious about stuff, including, I guess you're speaking about everything in the editors optimize, but, okay, you can always improve stuff. You can always find better plugins and macros and, "Oh, let me... You know what, this thing that took this pain point, I just found this tiny pain point, let me spend the next five days creating a plugin for my editor or whatever the fuck to remove that one pain point," when you should have just kept going as opposed to taking side quests.

ThePrimeagen

So, I have a rule which is I do not edit my RC other than some kind of cataclysmic thing, like someone updates a plugin, I didn't know they updated it and now there's like a hard error in my editor and I have to move forward. But I have a rule where I will edit my RC, my Neovim RC or anything once a year. Something that bothers me, I'll write it down, I'll remember it. I'll be like, "Okay, I want to change that," but I will just not go back to it. Now, every now and then I'll break that rule if I know, if like, "Oh, I want a new remap to be able to do this one command and that takes literally 13 seconds, like copy paste, do this, bop-bop-bop, done. Okay, I have this new remap, it made perfect sense in this situation, but I don't go plug-in exploring. I don't try to solve every problem. I don't want a perfect editor because that is a pursuit that will never stop. I just go, "This is good, good breakpoint, I won't do it again." So, last month I probably spent a hundred hours just editing every possible thing I could about how I start up my system and make... I can have a computer from zero to 60 in almost no time now everything the way I exactly want it, Neovim and everything all perfectly set up. Happy enough, I'm not going to touch that system again. Maybe I'll touch it next year. Maybe I'll take a year off. It's just I'm fine with that. I'm fine with not being perfect.

Lex Fridman

All right, zero to 60, let's talk about the perfect setup. What's your perfect programming setup, keyboard operating system, how many screens, chair?

ThePrimeagen

All right, I like all these.

Lex Fridman

IDE, let's go.

ThePrimeagen

So, keyboard, you're using my favorite keyboard right there, the Kinesis Advantage. Saved my career. Beautiful keyboard. Concavity and thumb clusters are just so important because if you really think about it, especially if you're using qwerty, when you're pressing the symbols on a standard keyboard, you're just doing this the whole time: backspace, enter, symbols. You're just doing this, and it just screws up your wrist constantly doing this. And this when you're constantly doing like ctrl and shift. And it just is like messing you up, so it's just like right here. That's so much nicer in life. So, keyboard most important, I'd say. Get that one done.

Lex Fridman

For people who don't know, Kinesis keyboard, I think the thing that you experience the most is exactly the thing you just said now, which is the backspace is really easy to press versus what it is on normal keyboards. So, backspace in general symbolizes you're deleting a thing, it symbolizes a mistake. Not symbolizes, it usually means a mistake. And so not only did you just make a mistake in what you were typing, you also have to take a physically painful action, annoying action to fix that mistake. And for most of us, we make a lot of mistakes, so Kinesis just makes it pleasant and fast and easy physically to correct the mistake. That's probably for me the number one reason of Kinesis. Everything else, yeah, super plus with the macros and the positioning, the concavity like you mentioned, but their mistakes are pleasant.

ThePrimeagen

Yeah. I am on that team, so that's why I love that. I would say that's one of the most important things. The next thing that I find to be very, very important is that one monitor. I'm a one monitor kind of guy.

Lex Fridman

What? Really?

ThePrimeagen

So, when I program, when I do anything... Now, when I stream, I obviously have a second computer that runs the stream because I sometimes crash my computer, I have to restart or whatever. So, I do have a second screen there that I put stuff up, but most of the time you'll notice that even when I'm streaming, you've been there, I have to physically switch to the streaming chat channel for me to read it, and that's because I'm operating off of one screen. And so I have this whole style in which I like to navigate, inspired by StarCraft, is that I believe in the press one key, go where you want to be mentality. And so everything about my setup is press one key. So, when I want to go to Twitch chat, alt-two, Twitch chat. When I want to go to my browser, alt-one. That's my browser. Alt-three, that's where I go to my programming. That's power finger, obviously. The big middle finger right there, just smash it down. Alt-six is going to be gimp, so my GNU image manipulation program, so if I want to draw, I go there. When I used to have Slack, it was alt-five. If I have a spare terminal where I need to run some extra things, that's alt-four. I had all these kind of... Everything is perfectly mapped out to single-key. And then when it comes down to using, say, Tmux, I have all my terminals into one single terminal. And now I'm able to kind of switch between there. Prefix one goes to my Vim editor. Whatever project I'm in, it's always the first Tmux tab, if you will. I'm not sure... They call it a session, but I'm not sure how to describe it if you're not familiar with Tmux. A tab. Second one is like my spare terminal, third one is my long-running process terminal, my fourth one is a long-running process terminal. So, I have it all set up, so every project I go to automatically spawns session one: Vim, session two: spare terminal, session three will also open it, so it's like everything's just ready to rock. Everything has been optimized to where I do that. If I want to go to a project, it's ctrl-F, and any terminal will bring up a fuzzy find list of every one of my folders on my operating system in which I can go to with just a couple keystrokes and, boom, I'm in that one now. And so it's very oriented to find where I need to be as quickly as possible.

Lex Fridman

Via keyboard.

ThePrimeagen

Via keyboard. Then in Vim I developed a plugin called Harpoon, which is I press one button and I can pin one of the files to like a temporary buffer. I think Projectile is potentially close to this in Emacs. I can't remember if Projectile... I think Projectile is closer to my sessionizing script. Anyways, so now I have four pinned files in which I can go to any of those pinned files with just a single keystroke. And so now it's just like... Because every time you develop a feature, usually you have like three files you're kind of primarily working in. And I can fuzzy find for the other files and that's that, but usually I just have like these three power files that I'm always swapping in between. And so it's like now everything is just, "I want go to the browser." That's one press. "I want to go to my workstation." That's one press. "I want to go to a specific folder, I need to change folders." Sometimes you work between two different projects, so in Tmux that's prefix, capital L will swap between your last two.

So, I have alternate projects, so I can even swap between projects in pretty much one key. So, it's just like do-do-do, just trying to optimize it, so I don't think as much, because I think search fatigue is a massive fail where you have to look for it. When I see people on a Mac do this and then explode all the different ones, that gives me anxiety. I'm like, "Why are you using your eyeballs to search for what you want to do?" Make it into a key press and never think about it again, ever.

Lex Fridman

You're making me think a lot whether I can live with your system, whether it's better because it feels better.

ThePrimeagen

It at least intellectually feels better. It may not be great for some people.

Lex Fridman

Well, there's a few profound things you said, which is like really what the number of windows or tasks you're switching between, whether it's programming, the number of files you're working on, it's small at any one time, at any one space of 20 minutes or something like that. So, okay, that's a profound truth. Sometimes we think like, "Oh I need the full freedom to search," but you don't. You usually work on a very small slice. But I guess the trade-off there... I always have three monitors, not when I'm traveling, but my happy place is three monitors. It's like, do you really need all of them to be present there? So, you're turning your head. Now, the monitors I have is two vertical ones, which is just better for certain kinds of content. They're positioned vertically, so you can read. You can use your eyes to scan quickly.

ThePrimeagen

Interesting. So, I don't even do that. I even have it so zoomed in that I probably only have like maybe 25 lines of code at any one time on my 27-inch monitor.

Lex Fridman

Yeah, I think that's... I think I feel fundamentally constrained when I can't see more because your eyes are just good at jumping. You could-

ThePrimeagen

Why not search? Why not press a couple of keystrokes? Ctrl-U, ctrl-D, jump up and down by a half page.

Lex Fridman

Because the ape visual system was designed to... You're loading a lot of information. If every time you have to investigate this table, what's on this table, you have to press a keystroke, you could develop the skillset that integrates that information but it's really... There is an

effective thing where if you have a sheet of paper like this and I'm looking at it, my eyes will be able to load in the structure of the information, the topics of the information. You just can do it faster, I think. There's a big cost because it's an extra monitor, but there is some stuff that's vertical when vertically positioned. See, code is an iffy one code because code, 25 lines at a time, I think you can do a lot. This is more for like articles and especially with visual information in them or documentation, you can just jump faster. But I'm trying to... As you were speaking so eloquently, I was like wondering, "Am I just like deceiving myself that I need that? Can I just keyboard shortcut-ify everything and just have everything on one monitor?" That's something I should probably try because I'm a big proponent of just automating everything with the keyboard because you can just move really, really fast, and you don't have to think. Because I also do creative stuff, whether it's recording music or video editing. It's hard... Some of these programs still make it super easy for you. On Windows, with AutoHotkey you can do quite a lot, but still there's limitations on how much you can do with the keyboard. So, it really is a pain in the to have to use the mouse, but, man, you're really making me think.

ThePrimeagen

Even the text one, the reading one, fundamentally I think I agree with you, that you can see a lot more and you can kind of look up and down, and see those two things. And probably in articles or things like that, if there's a graph down here that's really big that take up your whole screen plus text, I could see why that would be very beneficial to zoom out, to be able to have all that information, but for me, I can only look at like a square inch. Really, that's all my eyes can actually focus on. So, when I'm reading, I'm right here. Then I have to structurally try to pattern match what I think the information looks like. Then I have to start reading it. So, I'm not exactly sure if I actually get any real benefit of having a lot of stuff on screen, as opposed to I can relax my eyes so much I don't even have to focus. The words are so big. I actually program pretty zoomed in. My text is bigger than this when I program, and so it's just that it's so comfortable, I don't even have to exert any effort to read the code.

Lex Fridman

But you have to kind of train your brain to know that you can navigate spatially using keys.

ThePrimeagen

Yeah, Neovim by the way.

Lex Fridman

Oh, maybe it has everything to do with Neovim. Okay.

ThePrimeagen

All right. And then Neovim is obviously the next big one. I love Neovim. Reason being is that I think you can make all the arguments that you want about which editor is the best. I do not think you can make an argument that Vim motions aren't superior.

Lex Fridman

Here we go. Can you explain Vim motions? What is this? So, Neovim... Vim is an old school editor. Neovim-

ThePrimeagen

It's a modern take on an old school editor.

Lex Fridman

Yeah. And what's ELI5? What does it take to work with Neovim?

ThePrimeagen

Oh, okay. I thought you were talking about a Vim motion there. That's how... I know, but you know that meme that's just like, "Hey, Jarvis, can I tell you about Vim motions?" Because they can't fit anything else in their head because they only have Vim motions. You said EL5, explain it like I'm five, but in my head it's like, "Okay, E is jump to the end of the word, L is the one more..." Dude, I'm so broken that I'm like, "Okay, Vim motion," when I hear letters. Yeah, so, you can think of it like this is that Vim has a language to describe movements in text because its primary mode of operation is manipulating or editing text. So, it is a well-thought through set of movements, deleting, yanking, pasting, copying, all that kind of stuff that goes in, motions that are optimized for working with pretty much code. A good example, say you have three lines of code you want to delete. If you're in VS Code, take your little beautiful mouse, highlight those things, press the backspace. That's lovely. Your hand left the keyboard. Very simple to do though. It's very beginner friendly. I was a huge Vim hater, by the way, so I just want you to know that before we go into this. I was probably the biggest Vim hater. If there was Saul to Apostle Paul, I am like the Saul to Apostle Paul of Vim, just so you can see how big the gap was. Or you can do something that's like... I don't know what the VS Code shortcut is, but I'm sure there's some keys you can press to delete the current line you're on. Delete, delete, delete, you can just do that. In Vim, I can go DAP, delete around paragraph. All contiguous code in that thing. I'm going to delete, so D, then I can choose my motion I want to take, AP, around paragraph. Or maybe I want to DF, meaning jump up to the next character that matches the next character I'm going to press. So, DF opening parenthesis will delete everything from your cursor up to the first opening parenthesis. So, you get to describe your motion in these little keystrokes. And as you get really good... You've seen people that can master Fortnite, it's the same thing with mastering Vim motions. When you get so good, you no longer think about each individual movement, and instead you're just like, "Get rid of the paragraph, jump here, jump this, highlight this, yank this, do this," it becomes so fast that you can superiorly edit text at a very fast rate. And there comes a point when you know your language really well, you know the problem you're really working on really well, where editing text and getting code out actually becomes one of the many bottlenecks. People always talk about, "Well, most of the time I think..." Most of the time I'm not thinking, I'm programming. I know what I want to do, I want to go as fast as possible because I've been just doing it for so long and I'm so familiar

with the general space that it becomes a huge problem for me. I cannot tell you how many times that I've been purely bottlenecked by the fact that I just can't type fast enough and I just need to get it out of my head onto the text editor. And so that's why I think Vim motions are superior in all aspects. Keep your hands on the keyboard, on the home row, and it can manipulate text in very wide and fast ways.

Lex Fridman

Oh, so, this is not just about writing text, this is about modifying text. It's primarily about modifying text.

ThePrimeagen

Yes.

Lex Fridman

And I'm sure that most editors including Emacs, including VS Code can do all those same things, but there is something... They just don't encourage you to discover those things. That's like an important thing about a lot of technologies and programming languages that a lot of them can do a lot of the stuff, but it's something about whether it's the community or the style of the language or anything like this that encourages you to not be lazy in the beginning and learn the fast way to edit text, in this particular example, how to use the keyboard. That's a fascinating just reality of how technology is used. You want to be encouraged to find the fast thing as quickly as possible so that long term it's efficient and fun to use the thing.

ThePrimeagen

Yeah. It takes a long time for dividends, like a long time, but on top of that, notice I didn't say Vim. I'm not saying, "Go use Vim," I'm saying, "Vim motions." Let me give you one more example. I'm a big fan. Let's say you have a line that contains some variable, some function you're calling something that takes in a string. And you need to do that again, so you would typically copy that line, you'd paste that line below, you'd go into the string and you'd change the string. Let's say it's calling some sort of configuration, you need to call it three times with three different configuring strings. In Vim, I like to do shift-V to highlight the whole line, and then Y. Some people do YY, but I don't like to do double ones. I like to be able to do two different fingers because you can do that way faster than one finger twice. It's just a little optimization for me because you can't press that as fast. So, anyways, I'm very optimized in my approach, so I yank the line, paste the line. CI double quotes will delete everything inside the first occurring string. Then I can type the string, escape, save. And so it's like so optimized that I can just jump so fast in between that, whereas the copying and pasting line is probably the same speed, but the navigating to the string, deleting what's currently in the string, and then... That's such a fast motion in Vim, and I just do that all the time.

Lex Fridman

To backtrack, really dumb question, CI, what's the difference between typing the letters and using the letters to navigate and edit? How do you switch between the two modes?

ThePrimeagen

Okay, so insert mode means that you're just putting in text, and then normal mode means that you're moving your cursor.

Lex Fridman

And how do you switch between the two?

ThePrimeagen

Escape. Escape goes from insert mode into normal mode. And to go into insert mode press I to take your current cursor and go to the beginning, A to go to the end of the line Cursor, capital A to go to the end of the line, capital I to go to the beginning of line, O to put a new line below and then put your cursor at the proper intended for the language, shift-O to shift your current line down, and then put a new line in. You can see, there's a lot-

Lex Fridman

So, you're pressing escape a lot.

ThePrimeagen

Yeah, I mapped mine. I do ctrl-C. Ctrl-C does the same thing except for in one edge case. People hate that. I got used to it just due to the fact that I was using IntelliJ, and I really hate pressing the escape key, so I just got used to pressing escapee.

Lex Fridman

That seems like an essential thing to do if you're using Neovim to map escape to something.

ThePrimeagen

Cap lock would be your standard go-to.

Lex Fridman

Oh yeah, I map it too. Cool. I got you. I got you.

ThePrimeagen

Yeah, so then it's just really easy to press it, and boom, boom, boom, not a big deal at all. But yeah, I think that if you're willing to learn it, the emotions are superior, but if you're not willing to learn it, then they're not superior. You should just not do it. If you're willing to endure pain, it's good. If you're not, it's actually way worse. It's 100 worse.

Lex Fridman

Right, so if you like pain, you use Neovim. Totally. I understand.

ThePrimeagen

Yeah, you're totally on-board.

Lex Fridman

100%.

ThePrimeagen

See, now you get it.

Lex Fridman

If you like joy, you use Emacs.

ThePrimeagen

Sorry, sorry, did Emacs ever get a good text editor? I know they're a great operating system, but I never caught up if they got a good text editor.

Lex Fridman

Operating system? I think you've been miseducated my friend. So, at least 30 minutes on Emacs versus Neovim is what Reddit requested. Have you actually used Emacs in order to be able to talk so much shit or no?

ThePrimeagen

I used it for a year.

Lex Fridman

You used it for a year?

ThePrimeagen

Yeah, yeah. Doom Emacs, Spacemacs and regular Emacs.

Lex Fridman

But you don't even know Lisp, so did you really use it?

ThePrimeagen

I kind of hacked my way through kind of like, "Okay, so this is how to configure..." You can kind of get your way through and do all that.

Lex Fridman

So, you recommend to mastering Neovim and really learn the depths of it, but Emacs is okay to just kind of use before making a judgment. I think everybody...

ThePrimeagen

You got me on that one?

Lex Fridman

Yeah, no, and what's Neovim written? It's Lua?

ThePrimeagen

Yeah, so Lua would be the configuration language, but you have... It's written in C, but you have Lua 4. And Lua is just a dead simple language. Anyone can program Lua.

Lex Fridman

I actually don't know why... I think it's because my love for Lisp that I went with Emacs. I think you just choose a path and you walk down that path. And because there's just such a vibrant, intense battle between the two communities, you just start fighting just because everybody else is fighting. And then one day you're an old warrior on a horse, and you're wondering, "What was this all for?" And it's quite sad, in all seriousness, that I haven't to this day tried Neovim. I think because there is a learning curve. There's a learning curve to a lot of these editors.

ThePrimeagen

Yeah. To really learn it.

Lex Fridman

To really learn it. And I think this is some of the criticism of maybe VS Code or Sublime or Atom that it's so easy to not learn it, to just kind of half-ass use it. And there is a big benefit to having editors that force you to have some learning curve, where you take the art, the science, the procedure of editing seriously. Because you spend so much time in it, you might as well learn how to use the thing.

ThePrimeagen

My big takeaway really, what I'm trying to say with all these words is that I honestly don't actually think that... The editor obviously does not make the programmer, but I think it says a lot about your character as a programmer if you don't know how to use your editor well. There's something about a person who's willing to commit their life to programming, and spending literally 50,000 hours doing an activity over the course of their lifetime, and never take the time to learn their editor through and through. It just seems strange. You'd never see that in another world, where people would be able to build something or do something and just completely forget how these things work, and only just focus on one part of their

craft. And so, to me, it's just like it doesn't matter how you use it, I want to see the person that just knows how to use it, and they know how to use it well. When there's a problem, they can say why the problem exists, and then go and fix the problem. To me, that's like, "There you go. You've done it. You now know your tool, go forth and conquer with said tool."

Lex Fridman

Especially for tools you use a lot.

ThePrimeagen

[inaudible 03:46:07].

Lex Fridman

You have to look at your whole life, your life, whatever, if you're a developer or anything, what is the thing you do a lot?

ThePrimeagen

Meetings.

Lex Fridman

Yeah, yeah.

ThePrimeagen

Sorry. Keep going.

Lex Fridman

Ask a question like: how can this be done a lot better? Because every single day you do this for hours a day, how many hours did you spend on thinking how to do this better or whether to do it at all, in the case of meetings? People surprisingly just don't do this enough. I see this, just to go back to jujitsu, there's a lot of people that show up and do jujitsu or martial arts, and they do it the same way over and over and over, and they invest tremendous amount of energy. And they don't ask like, "How do I do it differently to improve faster?" In the case of jujitsu or any kind of sport, same with practicing the piano or the guitar, they just religiously put in a lot of time and derive a lot of joy from getting better. They don't enough ask the meta question of like, "How can I do this better?" And with editors, it's surprisingly how often people do just that. With typing, it's surprising how many people do just that. Like you said, they're pecking or looking down. It's like the quality of life improvement you can have by learning to touch type, by just like typing without looking. It's immeasurable. You're bringing a lot of joy to your life because all of us are typing a lot. And the reason, by the way, I was extremely efficient with Emacs... I'm sure you know, all jokes aside, it feels like Neovim has more room for the kind of efficiency I've had with Emacs to be able to move really fast as you described me to edit. There is a real joy. It's not just efficiency, it's a freedom that you can get when you get really good with an editor. The

reason I chose to go with VS Code is it felt like there's going to be an acceleration of features to which Neovim or Emacs will not be able to catch up, in the... and I don't mean in the next five years, I mean in the next 30 years. And it felt like I almost wanted to take the pain of learning new editors constantly and just switching and learning that, because I was getting so comfortable in Emacs, with this Kinesis keyboard, everything, all the shortcuts, I know how to program, and it felt like this is not... Neovim will not be here in 50 years. Possibly might be, I don't know, but it felt like you want to learn these constant different technologies. Cursor is a great example of that. I primarily am using Cursor now. I go back between VS Code and Cursor. Just the skill of using AI is a real skill, from the shortcuts to the timing to the layout of the windows to how I think about where, when and how to use the AI that doesn't distract me, that it empowers me, not just for the fuck of it or for the fun of it, for the actual measure of productivity. It's a skill. And I feel like I would be stuck in a local maximum of comfort if I stayed with Emacs. And maybe the same should be true for me with Neovim. I should try it seriously. I'm sure there's a plugin, like a copilot type of situation that you could set up with Neovim. I should possibly consider that. But Cursor is doing a lot of really fascinating stuff on the IDE side, not just sort of generate code and edit that code manually, it's like continuously be able to rewrite code. It's the idea of tab, tab, tab, tab, move the Cursor around, but also modify parts of code and do the diff really nicely, that whether it's Cursor or VS Code that wins that battle out with Copilot, I don't know. But that feels like a fundamentally different experience than the really efficient, joyful experience that you just described in your selling me on this is Neovim. That doesn't have an AI in the picture, obviously immediately, but you can, yeah, absolutely.

ThePrimeagen

I would 100% agree that Cursor seems like such a cool product. I actually think there's a lot of really neat things coming down with all of that. And I could change from Neovim. I don't use Neovim because I love Neovim, I use Neovim because I love the instrument I play. And so it's like if Cursor can meet those needs, I could see myself moving over. I don't have some sort of obsessed attachment with it. I am curious though that every time I use AI... I think I just have skill issues. I think I'm just so riddled with skill issues when it comes to using AI, I have yet- I think I'm just so riddled with skill issues when it comes to using AI, I have yet to be able to use it in a way that I really love it.

Lex Fridman

We'll talk about it, but before then-

ThePrimeagen

Oh, ball to sit on. I forgot to say that, ball to sit on. Desk needs to be properly heighted. One monitor. Eyes should be two-thirds way up the screen. I don't like to turn my head. I prefer my hands in a pistol neutral position. And there we go.

Lex Fridman

A ball to sit on. Yoga ball.

ThePrimeagen

Yoga ball.

Lex Fridman

What's that about?

ThePrimeagen

It just helps just maintain good posture, because when I have something to lean against, I do this.

Lex Fridman

You're for hours sitting without... Wait, what are you doing?

ThePrimeagen

I sit on the ball, and then I bounce.

Lex Fridman

Is your back leaning on a thing?

ThePrimeagen

No.

Lex Fridman

What the fuck?

ThePrimeagen

Well, how else do you-

Lex Fridman

You're the only person in the world sitting on a yoga ball as you program for hours. You do realize this, right?

ThePrimeagen

It feels great. The problem is whenever I get a back, I just slouch and I find myself just getting uncomfortable. And I'm like, "I'm uncomfortable." My shoulders are getting goofed up. I'm chicken necking constantly. It's just like-

Lex Fridman

But you're able to keep your posture for hours on the yoga ball?

ThePrimeagen

Yeah. And so I can just do that. And then I find myself, if I slouch, I'm like, "Okay, Nope. Got to get back.

Lex Fridman

Do you have incredible back muscles or what?

ThePrimeagen

No. Well, I don't think it takes incredible back muscles to-

Lex Fridman

Keep posture.

ThePrimeagen

... remain upright. Yeah, I think that's a pretty basic human function. I would not consider myself a strong person.

Lex Fridman

Yeah. Basic human function. I don't know.

ThePrimeagen

Facts and logic.

Lex Fridman

Okay, cool. With one screen. Neovim. What operating system?

ThePrimeagen

Linux, just because I want a good window manager. That's the whole press one button, bring up Chrome. I just use i3. I'm sure I could use something better than i3. People always tell me all these window managers are really great. But I just have those three screens I switch between, so it doesn't really... I don't really care what I use, just long as I can press one button and go.

Lex Fridman

Yeah, I'm the same, so half and half. Half Linux, the other half Windows with Linux, meaning WSL. What's that? Windows Subsystem for Linux.

ThePrimeagen

Weasel.

Lex Fridman

Weasel. See, no, there's got to be a better one that's more positive. Weasel just sounds -

ThePrimeagen

Seems right up Microsoft's alley. That seems perfect.

Lex Fridman

People often accuse me of being a shill for somebody, sometimes dictators. If I'm a shill for anybody, it's for Windows. There you go. I get paychecks every week from-

ThePrimeagen

Dang. Bought by Bill Gates.

Lex Fridman

Well, he's not Microsoft anymore.

ThePrimeagen

I know.

Lex Fridman

Developers, developers, develop. No, I'm just joking. I think, man, I need to try Mac. I need to try. I'm surrounded by people with iPhones. I use Android.

ThePrimeagen

I use the Android.

Lex Fridman

Yeah. There you go. See? Oh.

ThePrimeagen

We're losers together.

Lex Fridman

Losers on a sinking ship. Okay, just to stay on Neovim for a sec and to give love and a shout-out to your friend, Teej.

ThePrimeagen

He Streams, by the way.

Lex Fridman

He's a streamer. And I subscribed. And I've been enjoying it. My allegiance is slowly shifting from you to him. The quality is far superior with him, the looks, the intelligence, the skillset, everything, just far superior. No. Okay, he -

ThePrimeagen

You know you're making his day.

Lex Fridman

All right. He mentioned that he loves Neovim because it gives him the ability to eliminate having to do things he doesn't like. That's just a nice way to frame what this automation process that you described of automating a way, assigning shortcuts to things that are painful, that procedure. I wonder if you agree with that.

ThePrimeagen

Fully agree. We have very similar mentalities when it comes to usage of Neovim, why people should use it, all that kind of stuff, and how to even use it well. He definitely takes it probably to a further degree. He spends more time automating and all that. I don't necessarily derive a lot of joy from getting the perfect setup. But a lot to learn from. He's very, very good at what he does. He's 30 years old, been programming for not too many years, and he is one of the most talented developers for sure. It's very shocking to see how smart someone can be.

Lex Fridman

People should check him out @teej_dv. Teej.

ThePrimeagen

DV. His last name is DeVries. DeVries.

Lex Fridman

Oh, it's not developer. Okay, cool.

ThePrimeagen

Yeah, yeah, it's just TJ. That's just his name just spelled fun.

Lex Fridman

All right, Teej. What do you love about him?

ThePrimeagen

Wow. How much did he pay you to ask these questions?

Lex Fridman

Thousands of dollars. Thousands.

ThePrimeagen

Just so many dollars.

Lex Fridman

I can't even count that many dollars.

ThePrimeagen

Trust. Obviously trust is the biggest thing, especially in the, quote, unquote, "streaming" YouTube world, if you will. It's very easy to find people that will want to be a part of stuff. People tend to latch onto things, and it's very hard to find someone that you can really, really trust. And so he's just somebody whom I can genuinely trust. He will always tell the truth. He's all the right things for a good friend in this kind of endeavor.

Lex Fridman

As a good friend, he told me questions I could backstab you with.

ThePrimeagen

Okay, I hate him. I forgot how much I don't trust him.

Lex Fridman

Speaking of a harpoon, you mentioned it. He said to ask you basically how many years or decades it's going to take to transition to Harpoon 2 to actually release it, develop it, and so on. Can you describe what Harpoon is and why you seem to be incapable of finishing a single project?

ThePrimeagen

That was a lovely framed question. Harpoon 2 is actually done. This is what I did: To avoid the swirl in the thousands of questions I will inevitably get, I kept the master branch as Harpoon 1, and I've kept Harpoon 2 as Harpoon 2 branch. And people that don't read the read me to say that I just use Harpoon 2 now, that's their fault. That's it. I really don't like answering hundreds of questions about open source stuff. I used to love doing open source and all that, but I got my soul crushed during the Falcor years, and so I guess I'm just allergic to being a really active maintainer. I build everything just for me. Harpoon's just literally just built for me. I spent three months trying to figure out the most optimal navigation for files, and that's what I came up with. Harpoon, it's a take on alternate file. If you're familiar with the alternate file, typically you'll have this in all editors where you can go back to the file you were just in. And so that means you can have effectively two files you swap back and forth in. You've probably used it a bunch; really fast way to navigate. Pretty nice thing to do. I want alternate file, but three of them or four of them, and so that's all Harpoon is is just being able to pin a file. And so I have one button to press to go to a file, another for another, another for another. And so I can have up to four. I just had my four power fingers. For Dvorak, what is that? That's HTNS. If I go Ctrl H, T, N, or S, it goes to one of the four files. And that's it. That's all it is. And you can technically make it so you can add in functions and be able to execute things externally. You can open up terminals, you can send requests off to servers. You can do anything you want with it, I just have it primarily designed for opening files.

Lex Fridman

Since you mentioned it, what keyboard layout do you use? You use Dvorak?

ThePrimeagen

I use Dvorak, but I used a custom version of Dvorak. The reason why I used it is in 2017, we are just having my second kid, it was Christmas and I'm having so much pain in my arm and I'm sitting there freaking out like, "Oh my gosh, is this the end of my career? Am I done programming? Is this all over?" And so I decided that I was going to create my own keyboard layout optimized to prevent the pain that I'm experiencing, so I used to Dvorak as the base and then laid out the symbols in a symmetrical, reasonable way so that it's opening, closing, opening, closing, opening, closing. And they all are right here. I actually have to hold shift to press a number. Symbols are actually my first thing I get to press. And so it's very optimized for a laptop keyboard layout so I can use my laptop in a very efficient, nice way. That's how I got started on Dvorak and all that. I wouldn't actually recommend it because I didn't have a Kinesis at the time. I didn't even know Kinesis existed at that time. And so when I discovered Kinesis in also 2017, that's when I was like, "Oh, okay."

Lex Fridman

Would you recommend Kinesis to people?

ThePrimeagen

I'm technically sponsored by Kinesis, so it's hard for someone to believe someone that's sponsored by it. But I did use it before I ever became sponsored. They're the only sponsor that I reached out to and said, "I need a sponsorship from you. I'm going to use you either way. You can say no, but I really love it." And for the first three years of using Kinesis, they gave me free Kinesises, Kenisi, as my sponsorship.

Lex Fridman

Kenisi. Yeah, I'm always torn. I tried to leave so many times.

ThePrimeagen

You can't. It's too good.

Lex Fridman

But, see, I have this absurd situation of traveling with it.

ThePrimeagen

I relate.

Lex Fridman

Yeah. I'm literally going to the war zone in Ukraine, I have a Kinesis keyboard, a laptop, and just a few other small things and that's it. And it's like is Kinesis keyboard really going to be 30% of volume that you're bringing to a war zone? But-

ThePrimeagen

Looks like the answer is yes.

Lex Fridman

Yeah. Do you really derive that much value? I think it's probably spiritual or psychological for me. It feels like home. There's comfort associated with it. I try to leave.

ThePrimeagen

I love this experience. It's like a relationship you have with the thing.

Lex Fridman

It is. But I'm trying to figure out if it's a toxic relationship or not. I think it's mostly love. I think it's love. Like all relationship, there's some push and pull complications, but-

ThePrimeagen

They say that distance makes the heart grow fonder, so maybe sometimes the Kinesis keyboard needs to stay at home and the laptop keyboard can be the one so that your heart grows even more fond and that connection grows even deeper.

Lex Fridman

I already miss it as you say it, so I don't know. I think it's coming along to all the trips. If it breaks down, though... I was worried that Kinesis was shut down as a company. I'm like, what's the business model here? Who actually uses these keyboards? But apparently they're still going strong.

ThePrimeagen

Yeah. Who uses these keyboards? As you use the keyboard "I have to take it with me everywhere." I wonder who uses these keyboards.

Lex Fridman

Yeah. Yep. I should mention that one of the things when I first became a fan of yours, I heard you talk about coffee and term... I still don't, by the way, understand what you're even talking about. I need to actually use it. But you run, amongst many things, a coffee company. Man, this smells so good. This one is dark mode, dark roast, whole coffee beans. There is seg origin, dash, dash location. Brazil.

ThePrimeagen

Yeah, there's a bunch of stuff on there.

Lex Fridman

Stuff on there that's very devy. Shop, server, web. Can you legit, as such, order coffee via SSH?

ThePrimeagen

As of right now, it's the only way you can get the coffee is via SSH. Okay, can I just origin, origin story you?

Lex Fridman

Yeah, yeah. Yeah, right, I was going to do some kind of command line. Command to request or dash dash help or something or-

ThePrimeagen

Command coffee?

Lex Fridman

Command coffee.

ThePrimeagen

Okay. TJ and I, again, same Teej, Teej TV, about... By the way, very amazing designs done by David Hill. They're very, very good. Let me give the basic ideas. It must've been about a year and a half ago, TJ and I were talking like, "Hey, every one of these people that have some sort of following, some sort of online presence, they're always selling a thing," but I got nothing to sell. I don't really want to do merch. I've never really enjoyed doing merch. I just find that, I don't know, it's just not as much fun for me.

Lex Fridman

Don't want to have a tequila?

ThePrimeagen

I don't want a tequila. I want something that-

Lex Fridman

Like The Rock.

ThePrimeagen

And I also want something that I really don't feel bad about selling. There's a lot of people that will go on the internet and they'll shill for a whole bunch of products like, "Oh, okay, try this, try this." And this is why I've only ever really done Kinesis is because it's like, well, I can

point to something that was really bad in my life, I was very scared, and now it's not bad anymore. It's like, okay, that one made sense. But everything else always has been... It's harder for me. And so we just talked for so long, and we love Neovim, so we're just like, "Why haven't we could do something from Neovim?" And we're laughing about that, ordering from Neovim is just so ridiculous. And then at some point, we're just like, "Well, wait a second. And maybe we could do coffee. Every developer loves coffee. Maybe we could figure out this coffee business." And so I had a good friend named Dax, THDXR. Dax, yeah, Dax. The most sassiest man alive.

Lex Fridman

Sassiest?

ThePrimeagen

Oh yeah, he has a lot of sass.

Lex Fridman

Beard?

ThePrimeagen

Yep, he has a beard. He does SST. He does a lot of stuff. Very, very talented. We'll call him DevOps engineer. He's more than that. But very talented guy. Him and another person named Adamdotdev, vegan, by the way, great guy. We take him to Korean barbecue all the time. He eats nothing.

Lex Fridman

That's great.

ThePrimeagen

And Liz, she has been super important to the terminal coffee company. I think without her, we would not have been able to do what we have done. And then also David Hill, designer, he does Laravel. He designs for Laravel. Very talented designer. And so we all came together. And we were just laughing about how could we do something that's just ridiculous? And that's what we came up with. Yeah, there you go. You just open the website. You literally cannot order. We actually do not allow you to order.

Lex Fridman

The website is something that looks like the terminal. Use command below to order your delicious whole coffee bean. SSH terminal.shop.

ThePrimeagen

Yeah. You can only SSH into it. You have to copy that command and throw it in there. If you want to add in the little terminal shop for your known hosts, you could do that.

Lex Fridman

How do you handle payment?

ThePrimeagen

Through Stripe. And so one of the things, we'll be adding a mobile checkout to where I'll show a QR code in the terminal and you can just check out on your phone, but right now, you enter in your credentials, it goes to Stripe.

Lex Fridman

Via all terminal, like SSH.

ThePrimeagen

Yeah, SSH, obviously it stands for Secure Shell. It uses elliptical quantum safe algorithms to ensure that your data's not being intercepted.

Lex Fridman

Yeah, but does he use AI?

ThePrimeagen

I'm pretty sure Dax uses AI. That-

Lex Fridman

You said quantum, so I don't know.

ThePrimeagen

Quantum AI?

Lex Fridman

Can this-

ThePrimeagen

Fusion quantum AI?

Lex Fridman

Can this even be a company if it's not using AI?

ThePrimeagen

We have some crypto chains with some quantum AI that's powered by Fusion, so it's pretty wild. Anyway, yeah, we just came together where we thought, what is the... That was from the Mike Tyson fight. It was literally that night Mike Tyson kissed the reporter and then walked out without any clothes. We did an ad for somebody. But we decided to make a coffee shop, and then we thought instead of just making it Neovim, what if we made it from

SSH? Because everybody has SSH. You have VS Code. Launch VS Code. You can order coffee from within VS Code. Because your little bottom terminal that has access to SSH, bada bing, bada boom. It's fun. And so we really-

Lex Fridman

I love this.

ThePrimeagen

We just wanted to do something where there's no level and there's no world that makes me feel bad about selling this in people buying it. It's good ethical coffee. We developed the entire supply chain and everything. It's all packaged, it's all boutique. It's pretty high-end coffee. It tastes really, really good. At this point, I don't like drinking other coffee. I get upset about it because it's not as good. And so it's funny that I've fallen for my own stuff. I'm high on my own supply pretty hard right now. I just got done ordering 16 bags and gave it out to my family to try to convince them. But it's just something where it's like I didn't sell you a software product that's going to influence your startup that could potentially lead to disaster, I didn't convince you to do a bunch of stuff that's going to change your career, I just said, "Hey, here's some coffee." And it's like a fun experience.

Lex Fridman

Yeah, it's fun, everything. The humor on it is great. People should go to terminal.shop.

ThePrimeagen

[SSH terminal.shop](https://terminal.shop).

Lex Fridman

I'm speaking to people that don't know what SSH is. And there, you can read the command and then figure out how to use SSH in order to... It's a kind of documentation, right?

ThePrimeagen

Yeah.

Lex Fridman

On the website.

ThePrimeagen

If you can't use SSH, you probably should just not worry about buying our coffee. That's the whole-

Lex Fridman

Well, you can learn.

ThePrimeagen

You can learn. If you're active and you're a computer person, you'd like to launch the terminal and feel like a hacker, go for it. We even have subscriptions.

Lex Fridman

What I would love to see... This is how it came up I think on the cursor conversation, is that I would love it if an AI agent did this, like Anthropos computer use or something like that, actually took the action of ordering the coffee while it was programming.

ThePrimeagen

Yeah, like, "Hey, order me some coffee," and it actually go off. "Give me dark roast." Order coffee. It could actually go through the whole flow of order.

Lex Fridman

Yeah, the whole flow. But even better, if you didn't ask it to order coffee, you asked it to do something, and as a tangent, as a side quest it did that. Which is computer use does that. They showed off that it's able to go to I think Google for some images, take a pause, and then continue doing other stuff. Anyway, yeah, super cool idea. Love it. Speaking of which, let's talk AI.

ThePrimeagen

All right.

Lex Fridman

You've been both positive and negative on the role of AI in the whole programming software engineering experience. As it stands today, what do you think? What's your general view about AI? What is it effective at? What is it not so good at?

ThePrimeagen

Okay, my general view is it comes down to something that's pretty simple, which is that if you're doing something in which is very predictable, AI is really nice. When you're doing something that is just not predictable, AI is not very nice to use. If you're using anything that's more cutting edge, AI will not be using it, or AI won't be very good at doing stuff with it. It's not great at Zig because Zig is just, say, less documented. It's really great at TypeScript. I think there's a lot of interesting things that are going to come down through AI that I think a lot of people aren't really prepared for or thinking through. TJ's the genesis of this idea, but the idea that I think there's going to be a lot of market manipulation, if you will, through AI. Meaning, hey, you want to research, say, best woodworking tools. Someone's going to be buying an ad spot. Someone's going to be buying premium training data. They're the ones that get the big boosts in the LLMs. But LLMs don't really have to market as an advertisement because it's not really directly an advertisement, they just had a more premium spot, per se, in the training data; a little bit extra learning to it. It's like there's a lot

of things about AI that I fear upcoming. A lot of it just comes down to people not learning or making the trade-off where productivity is the only thing that matters. And I don't think productivity is the only thing that matters. If you want to build something complex and difficult, productivity is not the only thing. You actually are going to have to do deep learning and pursue it beyond the basics. And so I see AI as this really cool thing. It feels like a magic trick. I remember the first time I used it, I got early access to GitHub Copilot. In fact, Nat Friedman saw my Twitch clip of me asking GitHub for it, and he sent me early access himself. It was awesome. And when I used it, it predicted an if statement correct and my mind was just absolutely blown because I had nothing before then, and now it's just like first time ever. And I just remember thinking, man, this is going to change programming so much. And then the more I used it, the more I just... For me personally, I kept introducing bugs, and I couldn't figure out why. And what I realized is that I developed... I wasn't copiloting well, I was autopiloting much better. And my ability to read code versus my ability to critically think and write code, they're definitely different sets of skill levels. I don't consider as well when I just read code as opposed to what I write code. And so I struggled there.

Lex Fridman

I do think that's a skill set.

ThePrimeagen

Yeah. Skill issue for sure.

Lex Fridman

Skill issue. For people who are not aware, that's a hashtag thing sometimes used mockingly in this case. There's several layers mockingly, but also seriously, meaning the criticism is grounded in the fact that you lack the skill versus some kind of fundamental truth. Yes. I think that that's the reason I use actually Copilot cursor a lot is for developing the skill of editing AI so I can just learn how to do that better and better. Because I think as I do that better and better, I start to utilize AI better. At this time, it is a bit of a boilerplate code thing, but you can do out of the box novel design decisions or tricky design decisions from scratch but fill out stuff using AI and then just learn the skill of modifying. I personally just... It's more fun to program with AI. Even when I delete a lot of the code, it's more fun. It's less lonely. It's what I imagine pair programming to be. And I've never done it, but it just feels like that friction that you get when you're staring at an empty thing is not there. Empty function, empty class, it's just more fun, less lonely. And I do think that a lot of the easier type of coding, it really helps with like interacting with APIs, basic things that I would usually have to look up to stack overflow for. It's just really fast at that. As example, just interacting with the YouTube API. The YouTube API documentation is not very good. And you can just load it all in there and ask it to generate a set of functions that access the API, all kinds of read and write operations, and it figures it all out. Well, you do have to read. You have to read and check everything. And you start to develop the skill of understanding where it

misinterpreted the task. What is that skill? I don't even know. You have to be empathic about what the AI, what its limitations are. A lot of the times that has to do with prompt engineering. You have to at the same time understand what the AI is aware of. What did you actually give it as data to be able to generate the code? A lot of times, we don't realize that we're not giving it enough information. Okay, okay, all right. You have to be empathic. Be like, okay, these are the files it's aware of. This is the specifics of the question you asked it. You have to imagine you're an intern that doesn't know anything else. Oftentimes, we want the AI to just figure out the things that's left unspoken. But you can't know those things, you have to specify those things. And so you have to actually be much more deliberate and rigorous in the things you specify, is to spell it out. And so I just have this sea of prompts that I have saved up, and I'm building these library of different templates for prompts and it's a mess. And I'm sure there's a lot of developers that have this similar kind of mess. A lot of it has to do long-term with the tooling that's going to improve that. One, the systems are going to get much more intelligent when you don't need the nuance. And two, there's going to be the tooling that allows you to specify those things and load it in correctly and give all the context that the system needs in order to make the good decisions. And maybe the system asks you follow up questions with, "Here's things you didn't make clear," all that kind of stuff. A lot of that has to do with the interface, with the actual design of the tools. Like we said with Cursor, it's going to keep getting better and better and better. My sense is developers in general should be learning this to not be left behind, to see how that can be used as a superpower to boost their productivity, their effectiveness, their joy of programming versus be seen as a competitor to them or something like that. But for me already, it's been a big boost to productivity. If you measure the actual how quickly you're able to get a thing done, it's been a big... And measured not across minutes and hours but days also. Sometimes there's things I have to do that are not that important that I'll just out of procrastination will push off.

ThePrimeagen

I know that.

Lex Fridman

And AI helps me actually get it done, because that thing, the empty page, like I mentioned before, it helps me write the thing, get it done, get it tested, ship the thing. Maybe it's just because it's just less lonely to work with an AI. I don't know. I don't know if any of that made sense, but-

ThePrimeagen

It all made perfect sense. I really do like that phrase, it makes it less lonely. I think there's something to that that's interesting having just some level of interaction that's not just like an LSP autocomplete, having something that's actually a little bit more than just that where it actually is thinking through and you can see a different thought and you're like, "Oh wow, that's a way different approach than I would've taken. Hey, that's cool. I like these kind

things.” And the thing is that I’m not a AI negative person. I can see why people really, really like it. I used Copilot from when Nat gave me the access all the way up until about six months ago. I used it for quite some time. And I really did enjoy the things I used out of it. It did the opposite for me. I felt like I was more reviewing than writing and I felt like I was more just letting things slide where I just didn’t really think too heavily about stuff. And just I wasn’t as engaged. And so I’m like, “Okay, something’s kind wrong here.” And that’s just a me personal thing. I recognize that is not how someone should approach these things. That’s not a good reason for why you should or should not use AI. I just don’t think that that’s right. I could probably correct that and figure out a better way to do it. I’ve been meaning to have another AI round, and so I’ve been thinking about maybe I just need to spend two weeks in Cursor and just fully embrace what does it mean to be somebody like this? And what can I do with these new powers? Have they improved to the point where they’re actually good? And for me, because a lot of the decisions I make, a lot of the little functions I’m writing, it’s not because trying to write this function to solve this problem, it’s because I’m writing these functions or this set not just to solve this problem but because I know in about another 2,000 lines of code of building all these other things, I’m going to need to start doing this next activity. It’s like I’m trying to really try to chess move myself into the exact things that, as I let things go faster, I fall apart on that chess move. And again, skill issues for on my behalf. And I mean it in the truest sense of the word where it’s like I’m making a critique because I don’t use it well enough.

Lex Fridman

I don’t know if this is a general rule, this is my anecdote data. The better you’re programming, the less you want to use the AI, the more gets in the way. The good programmers–

ThePrimeagen

It’s fair enough, as far as I can tell.

Lex Fridman

The more beginner programmers are much more happy to use AI. When I use AI, it’s for basic for just... I don’t know if there’s a better term. It’s not boilerplate, but it’s pretty easy programming. And that kind of programming is much easier to do. The 10 X, not to use the meme, programmers that I know that are ultra productive and brilliant people, they hate AI. They’re like, “This is nowhere close to what’s needed.” There’s something to that. I still think they should be using AI just for the learning because it’s going to get smarter, it’s going to get better. It’s the same thing, it’s like when you super optimize Neovim or super optimize Emacs, you may not discover new things that are in the pipeline, so it’s always good to be training in that way.

ThePrimeagen

Let me ask you a question here just for my understanding. You talked about this idea that you have all these LLM prompts, all this big backlog of messy LLM prompts that you have these templates for that you can do various actions. You have these strategies of making itself explain itself and then do the right thing. As far as I can tell, that's really built into a lot of people. Well, then you make this phrase where you're like, but then at some point, the interface is going to get better, and maybe it can do a lot of these things better where I won't need that. Then my question is, well, is anyone actually falling behind for not using AI then? Because if the interface is going to change so greatly that all of your habits need to fundamentally change and it will be able to clarify and make all those statements, have I actually fallen behind at all? Or will the next gen actually just be so different from the current one that it's like, yeah, you're over there actually doing punch card AI right now. I'm going to come in at compiler time AI, so different that it's like what's a punch card?

Lex Fridman

Obviously open question. It's a fascinating one. I personally think, yes, you're falling behind. Not you, but if you're not-

ThePrimeagen

It could be me, it could be me.

Lex Fridman

... not playing with it, you're falling behind because the thing I'm doing with the prompts is you're learning, you're building up this intuition about how AI works. You're understanding what is its strengths and weaknesses? Not even the current version, but the next version and so on. What does it mean to teach an AI system about the world? What kind of information does it need to make effective decisions? I think that does transfer to smarter and smarter models. You'll need to make less rigorous and specific in details instructions over time, but you still have to have that kind of thing. I think it's a skill of almost empathy with an AI system because it doesn't know... You know what it's missing? It's missing common sense. It's missing long- term memory. A lot of things, when we talk to other humans, they have a basic common sense about reality, and AI systems often lack that kind of common sense. And they also don't remember things. You have to realize there's a constant blank slate happening. It's almost just a skill of talking to an AI system that I'm training. And by having to write all those prompts and communicating back and forth to understand what kind of prompts work better or not, you build up that intuition. And also just raw the skill of reading somebody else's code. Maybe for people who work on large teams, that's a skill that's already developed. For me, not so much, so learning how to modify the code that somebody else written is a real skill. And also, the other thing you mentioned, which is considering another perspective on a piece of code is really nice, but it is also a skill to understand, okay, this is what you did. There's a skill to asking a question that code that's been generated such that you can have a conversation about the approach that was

taken. I think there's just a lot of subtle, little skills involved in a cooperative endeavor to code, kind of like there was a real skill issue between you and Teej when you guys did the video of two idiots, one keyboard. People should go watch. You guys did the video of two idiots, one keyboard, People should go watch that video, where you guys obviously sucked at it.

ThePrimeagen

Yeah, co-using.

Lex Fridman

That was pretty cool, which you guys did, which is controlling one Neovim interface from two different keyboards.

ThePrimeagen

Yeah. And then we each get an allowance of certain characters or motions we could perform.

Lex Fridman

Yeah. And so you both had to communicate together. That's a real skill. I'm sure you can get super efficient with that, but it just takes time to learn that kind of thing. So yeah, I think there's some value to it, but I think there's a learning curve.

ThePrimeagen

One thing to be pretty clear is that I actually use AI quite a bit. I just don't use it for programming. And so one thing I've been trying to get it to is to be able to have a long interview or understand what Twitch Chat is saying and become Twitch chat and be able to speak as if it is Twitch chat. Try to learn how to prompt it in different ways. And so I think those things for me are just really fun. I tried to get it to learn how to play tower defense. I made a tower defense game in Zig and then made it play tower defense, and then played a Claude 3.5 against OpenAI. Claude 3.5 would do better during the daytimes, and OpenAI did better during the nighttimes. I don't know why, I have no idea what was going on there, but one would just start winning and the other one would start losing. It was just very strange. And so it's just this, I'm learning to prompt well, but I'm learning to prompt in a very different axis. I just don't find it very useful yet in programming.

Lex Fridman

In programming. And I should also say that I'm using it in every walk of life, in every context. I use that same kind of exploration about prompts and so on, I'm using and learning. I think it legit is a whole field in itself.

ThePrimeagen

Yeah.

Lex Fridman

Prompt engineering and how to interact with AI systems, I think it's worth the investment. Can you actually speak to that? Because I saw you're basically pulling from Twitch chat and having an LLM speak. I didn't realize, you're not reading the exact chat messages.

ThePrimeagen

Yeah.

Lex Fridman

You're doing kind of some kind of summarization?

ThePrimeagen

Yeah. So I ended up making like eight queries off to OpenAI where it's just like the first thing is like I have it have it like a default personality. "Hey, you're Randall, the manager, you're a software engineering manager." Kind of explain their position, what they like, what they don't like, and then be like, "These are the list of thoughts you have in your head and you need to talk to this person and ask them a question."

Lex Fridman

This is amazing.

ThePrimeagen

"Give me 10 of these responses that you think are probably thoughts that you have and you want to ask." Make it kind of give you a list and then be like, "Okay." Then re-prompt and be like, "Hey, you're Randall, you're this, this, this, this, this. You have these 10 questions before you and now you need to select one of them and reword it in a way that sounds more like you, the engineering manager." And I'm constantly trying to make it iterate on itself as opposed to just one-shotting it. And I found if I iterate too much, it loses what it was originally trying to ask if I don't do it enough and it's just too degenerate from Twitch chat. And so it's like I have a lot of improvement to do with this idea-

Lex Fridman

Just to clarify, you're feeding in Twitch chat, "You're a manager, these are the thoughts you have in your head, pick out some of the most profound thoughts"?

ThePrimeagen

Effectively. It's like depending on what I want it to do, I'm trying to work on a better system still for it.

Lex Fridman

Brilliant.

ThePrimeagen

And so it's like, "How can I give voice to Twitch chat? Can I make it so that I can create adversarial characters against Twitch chat or for Twitch chat? Can I incorporate YouTube?" All that kind of stuff. And how do you describe to an LLM to role-play into its position? And so just thinking through those kinds of things. So maybe I am having some prompt skills, but it's just not in the coding world yet.

Lex Fridman

Sure.

ThePrimeagen

One day I'll get there.

Lex Fridman

I saw that you were playing with different voices. There was like a sexy voice?

ThePrimeagen

That started off as a French voice-

Lex Fridman

French voice?

ThePrimeagen

... and then it turns out ElevenLabs just cannot do a French lady. And when you do multilingual French lady, she starts talking. It's like, "What? What is this?"

Lex Fridman

I tuned into one of your streams and there's this lady in a sexualized way.

ThePrimeagen

It became too funny. And so we call her Not French Stormy Daniels.

Lex Fridman

Oh, nice.

ThePrimeagen

Yeah. But I want to go back to the AI and some of the aspects.

Lex Fridman

Sure.

ThePrimeagen

And so my big gripe with AI has nothing to do with its capabilities. It's exactly capable, as it should be capable, because that's what people programmed it as. The things that I really dislike is, A, there's a whole group of people that are just like, "The end is nigh. AI is here, you just need to stop programming." I cannot tell you, even you mentioned Peter levels earlier, he made some sort of tweet and one of the person's responses was, "No one in 2025 or whatever should be acquiring hard skills. You should rely on everything for the AI effectively." And it's just like these are really damning pieces of advice for young people. Young people are being told that you should never become an expert in anything, you should always offload. And the problem is that anyone worth any of their salt will tell you that AI, though can produce code, is going to get it wrong in a huge number of cases. And as the code becomes bigger or more complex or more input, it's going to just start kind of sloshing back and forth between bugs. And so if you don't have those hard skills and you're not ultimately the driver at the end of the day, you're going to really find some hard times, and your ability to progress will be directly bound to how good the LLMs are. So if you believe that the LLMs will be vastly superior to humans in the next year, maybe that's a good bet. But if they aren't, then your skill ceiling is bound to whatever they are. And even beyond that, there's just a level of information problem, which is like, "Do we even have enough compute power to be able to solve things at this real scale?" And even if we did, if everybody started using it right now, "Do we even have the compute power for everybody to use it right now?" There's a lot of kind of bounding questions, there's privacy concerns, and I just don't want people to make the immediate, or what appears to be the obvious choice, where you don't need hard skills, you don't need these things, we just need to only think creatively. It's like, no, I don't think so. I think these hard skills are going to be around for quite some time even with a massive improvement in the AI, you're going to really be needed to step in regularly for quite some time as far as I can tell.

Lex Fridman

But I also think even on top of that, just even acquiring the hard skills or whether that means, programming from scratch, for example, in the context of programming, that's going to make you better at steering the AI, not just correcting the AI, but steering the AI. I think there is some kind of, if you know how a computer works, you can program Python better. It's maybe counterintuitive, but if you know the low level abstractions, some intuition around that, you can steer the high level abstractions better.

ThePrimeagen

Yeah.

Lex Fridman

But that just seems to be the case. Unless of course AI becomes like truly super intelligent like many levels above, but it's very unlikely in the short term. And in the long term it's still good as it gets better and better and better to be able to ride the wave of the improvement.

ThePrimeagen

Yeah, I'm on that team very much so.

Lex Fridman

A lot of people have written to me, I think a lot of developers, programmers are really concerned about the future of their profession in the context of quickly improving AI systems. So do you think AI will eventually replace programmers?

ThePrimeagen

The hard part about that phrase is you use the term eventually, meaning do I think in five years, 10 years, a hundred years? What does that term actually mean? I think at some point if all things continue at the current rate of improvement, there does come a point where programming as a hard skill does become unnecessary. At some eventual point, way, way down the road, yes. I don't know what that point looks like. I don't know when it's going to happen. I don't even attempt to make predictions about that. But there are still some leaps and bounds we need to make. I mean even just societally, there's plenty of companies that don't even allow you to use AI. I mean, there's just practical problems that exist. So that's a question I just try not to answer in the direct sense. There will come a day if humanity continues and all things continue in a good positive direction, where a lot of skills will go out the window due to immense computing systems. So, yeah, I'll give you that one. But it's just like if I don't think it has anything in the near term, there's been no computer improvement up to this date that did not result in more jobs.

Lex Fridman

Yeah, absolutely. I would say that I think it depends how you define programming also, because when the community moves from assembly to C, from C to, I don't know, Python and JavaScript, that's evolution. That's really painful for a lot of people who are used to programming that lower-level language, so there's going to be a continuous evolution. And maybe that means with AI, there's going to be more and more evolution towards natural language as part of the tool chain like being able to learn how to write proper prompts. Yeah, because natural language is still a language. And in the long term, it's possible that a large percentage of programming is natural language. There are probably still going to be some percentage that's not, that's going to be extremely structured language.

ThePrimeagen

Right now, I don't think we are anywhere near natural language being possible because it's ambiguous. And I think what we'll end up seeing as people push really hard into this, you're going to see some sort of pseudo-lang, which is going to be a language for AIs in which you prompt, which is going to be less ambiguous. People keep striving towards the less ambiguous state. And at that point you're just programming yet another evolution into a higher order language. And perhaps that is a future in which people will have a more terse language. I'm just not sure how much more terse it can get. I mean, all I see is that if you say

natural language can be used in the pipeline, you've just made that many more people can become programmers, which means that much more software will eventually be created, which means there's that much more software that will need to be maintained, and just becomes a real big snowballing effect.

Lex Fridman

But there's just people who are programmers who are worried about their jobs. Not a complete replacement, but maybe a rapid evolution of what it means to be a programmer. Like you mentioned, if natural language becomes a way that you can communicate or you can program, that means the pool of people who can get programming jobs changes rapidly, so they're really concerned.

ThePrimeagen

And to some extent, because no matter how much we want to say how good AI is, there comes a point where there exists a bug, there exists a large piece of software in which to describe the change requires just pages and pages of description to the point where it is significantly just faster or easier for someone to just whip something out. There's definitely a balance there, it's not like a perfect trade-off. And so I think people need to quit worrying and think about how they can integrate it and try, like prove it to themselves. Do they actually make themselves irrelevant? And if you truly make yourself irrelevant, I would challenge you that you're just doing something that was just slightly too complicated to automate. If you're only writing just straight up crud apps from backend to front end and simple table displays, yeah, maybe we just couldn't quite automate that away and now we just have something that can just do that a little bit better, so now that's automated away, but that's not really programming. That's almost like building Legos at that point, where the designs are already set, you just simply have to move piece from bag into correct position.

Lex Fridman

Yeah. Is there something you recommend how a developer or programmer could avoid the situation where AI can automate them away?

ThePrimeagen

I think that the bigger the project you can manage, the bigger the thing you can build, the more understanding both down and up the stack you can go, the more valuable you become. Because if you understand how to build something in the front end, okay, well, now you kick off some LLM task of some sort, that's going to go off and make a change to the front end. Okay, while it's doing that, you can go and kick off something in the CLI tool, you can go and you can go kick off something somewhere else. And as these things come back with results, you can review the results, make sure it's the way you want it, change it, commit it, go to the next. You only become more, as you said in the end, more productive if we reach this state where it's truly able to do that.

Lex Fridman

I think there is like a skill to working together with AI, which is why I'm kind of excited to watch you keep trying to do it.

ThePrimeagen

Yeah.

Lex Fridman

It's like we don't know how it fits exactly, but it feels like AI should be a boost to productivity. And I definitely think it's a boost to just the joy of programming. I think there's a lot of people, yeah, it's a job, but it's also a source of meaning, a source of joy. Programming is fun, you're creating something cool, and also potentially that a lot of people use.

ThePrimeagen

There's this one thing that just really frustrates me, and this is kind of going into the Devin category, which is that I want an intern that cares.

Lex Fridman

Yeah.

ThePrimeagen

You don't get that out of an LLM. It does not care, meaning that I don't want it just to make a UI for me that displays these icons like I asked, I want it to care, I want it to think about it, I want it to present to me and me being like, "Oh yeah, yeah, that's great." And then me to make changes. And then later on it's like, "Actually, I really rethought about this and actually it'd be way better if we change..." It doesn't actually care about the craft. But when you work with an intern or you work with somebody else, they care. When they factor something, they actually go over and go, "Oh yeah, this is actually kind of bad. I'm going to come back to that." They finish this, they go back over here and they make this even better. They actually care about the thing itself. It's a completely different experience. I just want something that also cares that wants to make the thing better, not just simply accomplish the task. And I know I'm asking way too much that's not... Now we're getting into Blade Runner's level AI. I just want something that it just feels like I'm missing that, where it's just like it will complete the task to whatever level it understood what I was prompting, but it doesn't actually care about it.

Lex Fridman

I mean, there's so many aspects to caring versus the trivial version of that is a kind of restlessness where you want to keep improving, and I think that is very much AI could do, where it constantly just ask itself, "Can I make this better?" And if it keeps doing that, it probably is going to take it to some ridiculous place, so actually it's also knowing when to stop. I think developing something you can call taste, which is like trying, working extremely

hard, constantly improving until it just feels right. This is it. And I think that is a thing that AI is not good at where it's just like, "Yes, this is it."

ThePrimeagen

I've had write iterated three times and three was the-

Lex Fridman

Yeah, that's it. We're now there. And I think ultimately that is what humans are amazing at, which is like knowing when something is right like, "This is it." Especially as you understand, as you develop taste in a particular industry, in a particular context application, knowing like, "This is it. Yeah, the rounded corners on this button, that's exactly that. That's beautiful." So it's just a sense of beauty, a sense of function, and efficiency, and so on. Yeah, but humans could do almost like supervision of AI systems in that context.

ThePrimeagen

Yeah.

Lex Fridman

Yeah. You've ranted about Devin just full of rage.

ThePrimeagen

I mean, first off, the people that run Devin are extremely nice. I want that to be understood. I don't have some sort of upsetness against them or anything like that. Second, Devin, it's like the full package when it comes to programming. So it's going to have, you're going to give it a task and a repo, and it's going to go through, it's going to try to understand the repo and the task, make the change to the repo by exploring it, then actually make a commit to GitHub and explain what it did. So hopefully you have this whole offline thing, which is the other part of this AI part that I actually really like, where it's just like, "Go fix this thing." Then I can just go and unbroke and fix this one thing and come back and go, "Okay, good enough, merge, boom." I want that kind of running, being able to complete things. I think the ideal solution is that you can start giving it small bugs and it goes and fixes these bugs and you can just come back to these backlog tickets that no one ever does, and it actually starts going through these backlog tickets, and it's actually a really amazing experience. So I love the idea. I think we can all agree that that sounds great, but every time I've done it and I've asked it for many and I try to keep narrowing down the problems, the more narrow the problem, the better it does. So if I'm like, "Just add one singular icon. And when it gets clicked, I want you to do this just console, click me. At least create me an SVG and place it so it's nicely placed." The more narrow the task, the more likely it's to be successful. There's like a certain level of specifying where if you specify too much, it just like can't do it. If you specify too little, it just does weird things. So it's kind of like this very kind of fun, unique way you have to play the balance game. But so far, every time I do these things, I always end up going, "Gosh, I should just get better at Tailwind and write it myself," because I always go

back and I just rewrite it, and then it's just like, "Dang it. What am I saving at the end?" I feel like I'm not saving anything yet. And it's just like this, "I want it so bad." I actually want AI to be great because then I can really go fast. I mean, I can go amazing fast, but then I always just go, "Gosh, I should have just learned Tailwind myself to like the nth degree and just go fast."

Lex Fridman

Yeah. We should also mention that debugging, this might be intuitive or counterintuitive, AI is really bad at.

ThePrimeagen

Yeah.

Lex Fridman

Like that is one of the hardest... It actually makes you realize how special humans are and how difficult the task of debugging is. Obviously, for trivial debugging, maybe you can find bugs, but that is the real art of programming is finding bugs, logical bugs, extremely complicated rare bugs, edge cases. AI can assist, but man, the hard ones really require so much context, so much experience, so much intuition from, again, operating in a fog full of uncertainty. It's hard.

ThePrimeagen

Yeah.

Lex Fridman

Of course AI could maybe create logs and do traces and do some kind of loading a huge amount of data that humans can't, but ultimately that just means it could be a better assistant in debugging versus the actual lead debugger.

ThePrimeagen

Yeah. I mean, it'd be great if they could. I mean, the more it can do that, the better, because as far as I can tell, correct me where I'm wrong on this, current state debugging is really, it looks at the code, it looks at the bug problem, it just kind of tries to text-predict where it's most likely accurate, and then just tries to fix that spot. And so it's like it's likely this spot, you said admin panel, it's slightly off, this, this, this. It's probably this location, which could actually be a really great way to do search, let me do semantic searching, point to me where this is, because maybe that is a really great way to navigate large code bases. It's like smart intelligent search. As opposed to trying to make it do the thing, ask it to just help you do the thing in pinpointing problems. I'd love to see more of that, because that's for me is like the exciting part. And there's this really great article by creator or maintainer of curl, it's the I in the LLM stands for intelligence. And he writes curl and maintains curl. Curl has been inundated with security problems and all this, and it's all from LLMs being like, "Oh, I found a

security flaw. Here's the security flaw," details it out in the code. And he's just like, "Okay, how did you reproduce that? Show me," because if you look at the code right here, that's actually an impossible situation you're speaking of. And it's just like going in these circles and security right now is being inundated, these bug bounty programs are being inundated by LLM-submitted responses because they can't actually analyze the code beyond just like basic text prediction. "Oh, this is a stir copy. Stir copy is commonly referred, blah, blah, blah, blah. Boom, there you go. Here's the bug." And it's just like, "No, that's actually impossible because the if statement right beforehand leaves the function if the string is too long, so it's like we don't even run into this case. It's impossible what you're saying." So debugging is very interesting.

Lex Fridman

Yeah. I mean, that for me would be the big, if it can solve that, not solve that, but improve that, that would be huge, whether it's agents or just LLMs integrated into IDE.

ThePrimeagen

I think there's this whole idea, I call it a denial of attention. I think there's an entire attack vector that's going to be happening. We're using LLMs to generate fake bug reports, fake all these things to just actually effectively to demotivate and hurt open source maintainers. Polykill was the first bug that kind of had this experience, is this denial of attention where an active malicious maintainer just hounded the owner. And then a white knight came out and offered to buy some stuff from under them. And when they bought it, they actually replaced it with a malicious piece of code and then used it. So there's this whole security world that's developing around using these in a very aggressive format.

Lex Fridman

I mean, it's a fascinating world we're entering into, but I do agree with you that human developers will be a huge part of that world.

ThePrimeagen

Yeah.

Lex Fridman

That the job might evolve, but it's going to be there. If I can, I didn't really look at this page, I thought it would be cool to go over with you. This is, again, the-

ThePrimeagen

Stack overflow, my favorite

Lex Fridman

... Stack Overflow Developer Survey, talking about their sentiment and usage of AI systems. The general sentiment of, yes, 61% say yes, they use it and 25% say no, don't plan to. So

majority use it, majority have a favorable sentiment over it, favorable or very favorable or indifferent. That's like looks like over 90%.

ThePrimeagen

That's really surprising that that many people just have no plan in looking into AI. As much as I don't like using it for coding, I hope one day I can use it more. And so it's like, to me, I'm always looking for the next thing. I'm just surprised that people are, that, I guess obstinate for it. Obviously, the second one, the AI tool sentiment, it must be only the users who responded to the top two of that first one just given the amount of respondents.

Lex Fridman

I wonder if no and don't plan to are people who have tried it and quickly built up the intuition like, "This really sucks."

ThePrimeagen

Yeah.

Lex Fridman

So it could be like experienced programmers. They're like, "No, this is not making me more productive." 81% agree that increasing productivity is the biggest benefit that the developers identify for AI tools. Okay, so this is, what are the benefits? Increased productivity, speed up learning, greater efficiency, improve accuracy in coding, make workload more manageable, improve collaborate. Where's the fun, increased fun? I would say that's like number one for me.

ThePrimeagen

Maybe speed up learning is like a subcategory of fun. If you're able to learn more and be able to become better. To me, that sounds good.

Lex Fridman

I don't know. It's different because productivity is part of fun too. There is just a lightness. I mean, maybe improved collaboration, all of these elements for sure.

ThePrimeagen

My time using Copilot, there was certainly a level of wonder that would happen for quite some time where it's just like, it's just amazing what it can do.

Lex Fridman

Yeah.

ThePrimeagen

I'm just super impressed by what it can do, even though I don't use it. It's amazing to me that we have something that can even get that close.

Lex Fridman

In terms of accuracy of AI tools, only 2.7% highly trust-

ThePrimeagen

I would say that you have to be very green to think that you should highly trust an AI output. You should be very skeptical.

Lex Fridman

Yeah, I don't know where I stand. Probably somewhat distrust. Highly distrust seems aggressive.

ThePrimeagen

It does seem a little, true. You should always assume that there's something wrong, and then from there you can go and challenge it.

Lex Fridman

And then estimation of whether AI can handle complex tasks, most people don't think it can handle complex tasks. I mean, it seems like people have a good sense of what it's able to handle and not.

ThePrimeagen

I would argue that people don't have a good grasp of what complex is in programming.

Lex Fridman

Sure, yeah.

ThePrimeagen

If you say, "Write me quicksort," some people will think quicksort's super complex. But I would argue that that's actually probably the simplest thing you could ask an AI to do. Things that are so well documented, it's going to do a great job at that.

Lex Fridman

Yeah. Probably high-level design decisions, which people don't even use AI for right now, I guess agents are supposed to be doing that kind of stuff. That's probably the most difficult thing or the most impactful thing, or the most difficult thing is finding bugs.

ThePrimeagen

Yeah.

Lex Fridman

AI tools next year, writing code, and so on.

ThePrimeagen

Now, this one, the ethics part. I'm actually super curious your take on the ethics. Will we see Europe laying down some new regulations?

Lex Fridman

Oh, boy.

ThePrimeagen

What about artists? What about people that are really... Because the difference between coding and artists is very, very simple. If you gave me a sheet of paper, I could draw you a crab. You go, "That's a crab." But you can't do that with coding. It's like it's right or it's wrong. There's not a variation of interpretation for what a crab is. It's like, "No, you cannot make that statement." It's very bounded in what it can express. And I could see why artists, that's a very frustrating point. And then who gets rewarded for all that? And then there's like the whole thing with coding and licenses. How much of it is GPL licenses, do you think, they've scraped and used as training data? GPL forces open source. What are you going to do with that one? That means your model might need to be open source. OpenAI may have to get forced open, all their previous stuff if there's any hint of GPL.

Lex Fridman

Yeah, that's a weird one. That's a really weird one because most of these models I think are training on data they don't technically have rights to be training on.

ThePrimeagen

Yeah, there's a lot of questions.

Lex Fridman

There's an unspoken, it's a real Wild West.

ThePrimeagen

Because you could imagine that, I always use Europe because they tend to have like maybe the most consumer protection laws out there. You could imagine what happens if a law came down that said that if you used a model that produced GPL potential code, you have to open source? How many companies are going to be like, "Oh my gosh"? Like, "You have one year to get rid of all code that was generated that's potentially GPL-sourced from a model." You could imagine just the sheer panic that's going to happen. It'd be a fire sale of code.

Lex Fridman

So given all that, can you give advice to young programmers? This is another question from Reddit, the infinite wisdom of Reddit. "What should a person in their early 20s do to move forward in the tech industry?" And this is an interesting addition to the question, "And by doing it, will this be walking on someone else's path?"

ThePrimeagen

I am going to try to answer that question, I guess the best I can, which I think that if you're entering into the tech world, one of the hardest pieces of advice that I took a long time to learn was I became enamored and addicted. Obviously, we talked about that I program for way too many hours, forgetting to spend the time I needed with my wife, with my friends, all that stuff, like totally wrapping myself up into one activity. I think though it made me who I am, it was probably an unhealthy activity and probably not a wise activity. And so the best advice I can give is that you got to develop the love, the skill, the desire for it. Whether that's just only using AI agents, programming yourself, using Zig or programming JavaScript, whatever that flavor is that's going to get you coming back every single day, getting the reps in the gym, if you will for programming. But also knowing how to value what is valuable and not getting lost in the sauce where you're just so stuck on trying to make the next greatest startup that you sacrifice your health, you sacrifice your relationships. Or even worse, you sacrifice your own morals to take certain shortcuts that you probably shouldn't be taking in life to be able to achieve these things, because I'm sure there's hundreds of horror stories you could hear where people definitely shortcuted their morals for monetary success.

Lex Fridman

Yeah. I mean, the golden handcuffs comfort can destroy the soul in some sense. Yeah, so that's really important to remember. There's young people kind of thinking, "Do I even want to be a programmer now?" It seems like AI is getting better and better at programming. If they were trying to make that decision, would you still say, "Yeah, if this is something that fills you with joy"?

ThePrimeagen

I still want my kids to learn how to program if I can answer that, if that's a good enough answer -

Lex Fridman

Yeah, that's a really powerful answer.

ThePrimeagen

... in the sense that my kids are decades younger than a young person trying to learn how to program right now. And so I'm hoping that my kid can run and build whatever he wants in Roblox. I'm showing him ChatGPT and be like, "All right, let's ask questions. How do we do this?" It's still extremely confusing for him to do all these things. And so it's like, "Let's do

this." I want him to learn and be effective, and maybe one day he has to throw away all those skills in 20 years. But I bet you that whatever skills he threw away or whatever hard skills he had to throw away, an entirely new field that none of us have thought about, just like if you would have asked somebody in the '70s about social networks, they'd be like, "What the heck are you even talking about?" Things will exist in the future that are going to be massively different, and crazy, and exciting.

Lex Fridman

Maybe in virtual reality.

ThePrimeagen

There you go.

Lex Fridman

Maybe all of us actually down the line will just be building video games.

ThePrimeagen

Just entertainment for all, the brave new world of our world?

Lex Fridman

Well, I think entertainment is a kind of trivialized version of what a video game could be. It's like, what is the purpose of life anyway? I mean, it could be a deeply fulfilling video game. It doesn't have to be just like dopamine rush. It could be educational, it could be scary, it could be challenging, forcing an evolution, the leap into adventure that it makes up a fulfilling life. That could be video games. Who knows? Especially in virtual reality. I tend to... That's the other thing. I play a lot of video games. I think there's a lot of room to make video games deeply fulfilling, like there's a lot of space where that can go.

ThePrimeagen

I didn't know you played a lot of video games, because when I asked you specifically, "Should I play World of Warcraft or do Advent of Code," you're like, "Advent of Code, Advent of Code."

Lex Fridman

Oh, well, that might mean I've never played World of Warcraft because there's certain games I avoid. Fortnite, by the way, I think was one of them because I was worried it'd become too addicted.

ThePrimeagen

Yeah, yeah.

Lex Fridman

So there's certain games I just know I won't get super addicted to. Like for example, I'm terrified of Civilization. I have never played a Civ's game because I'm worried. I'm worried the dark path in my lead because there's some games just really pull you in. I'm much better with, that's why I play Skyrim. I can play these games or a Baldur's Gate, and moderate how much I play. And they could be like a lifelong companion versus an addiction where it's like sunrise and you're like, "What's happening with my life?" And I find myself naked behind a dumpster somewhere just wondering what happened. Yeah, so that's how I choose my video games.

ThePrimeagen

You're not the first person who has specifically called out Civilization.

Lex Fridman

Yeah.

ThePrimeagen

I've had more than one person, also very high up in the tech world, be like, "Civilization is my downfall. If I get near that game, I'm done."

Lex Fridman

Yeah.

ThePrimeagen

I've never even played the game. Now it makes me be like, "Dude, I got to give this a try. That sounds crazy."

Lex Fridman

Yeah. And the new one is actually supposed to be-

ThePrimeagen

Give this a try. That sounds crazy.

Lex Fridman

Yeah. And the new one is actually supposed to be really, really good. What were we talking about? Yes. For that same young developer, is there a trajectory through jobs that you could give advice on? So you started out with Schedulicity?

ThePrimeagen

Yeah, that was my first full-time.... When I had the government contracting one before, that wasn't quite full-time. It was in C. It was a lot of fun. And then building my own startup for

quite some time. So if you count either of those as full-time, then those would be the full-time. Schedulicity was the official on the docs.

Lex Fridman

Is there some value to jumping around, working in one company and another to try to figure out what brings you joy?

ThePrimeagen

I think there's a lot to that because not every job you're going to get is going to be great. Now, your first job you could get could make you think you hate programming. It happened, I did an internship at a place and I keep on surprising you with more kind of things I did in the past, did an internship at-

Lex Fridman

Fuck. You did so many things. It's incredible.

ThePrimeagen

... a place called Total Information Management System. Remember when I talked about that hours ago, about healthcare and that and industrial shipping and all that? It was a C-sharp shop. It was so bad that after I did that, I went and changed my major to mechanical engineering first semester in college.

Lex Fridman

Oh, boy. Oh, boy.

ThePrimeagen

I thought I, "Okay. Actually I like computer science. I hate the programming." So just because you've had a job doesn't mean it's going to be the one. And the thing is, here's the best part though, if you get a job and you like it and you want to do it and it's exciting, you don't need to change. I think a lot of people are like, "Oh, I got to find the next thing. I've been here for two years." There's of this, you got to move around mindset. I don't think you have to move around. I don't think it hurts your career. Because if anything, you'll gain more responsibility and you'll be able to talk with way more authority. And the next time you interview, you're going to be way more into like, "Oh yeah, I had to get these X people and these X people to be able to do all this stuff." And it's like you can talk with much more authority if you stay at a place longer. And that's nothing but benefits in my book. It's only if you stay at a place because you're afraid or you don't want to... You already have something that works for you and you just never want to change and you're just like, "I get to go in and just be completely mindless." I think if you go mindless for a couple years, you'll find yourself... That's the only real danger. You just come out with nothing at all.

Lex Fridman

Especially when you're younger, that's the whole point. You're like, "Take the risk. Take the leap out to the next thing, to the next thing." And not for money, but for just personal joy, joy.

ThePrimeagen

And money could get at the end, that's the best part. When you don't strive for the money, sometimes the money just shows up anyways.

Lex Fridman

Yeah. And some of the, what makes life worth living is the people you work with, a good team. Some of it's not to be generic, but culture matters. It's whatever makes you happy. For example, I just had won't call out places, but there's certain companies where everybody is very nine to five. And even if the work is exciting, they don't work hard enough I would say. I'm one of those people that likes to go all out, likes to be surrounded by people who are super passionate. Now to be fair, a lot of them don't have families or don't-

ThePrimeagen

Yeah.

Lex Fridman

It's a fascinating choice. I really don't want to talk down on any choice like work-life balance or not, I think both are beautiful paths. And if you really derive a lot of value from joy from your work, going all in, at least for some stretch of your life is a beautiful thing to do. Just all out, full-on passion, sacrifice a lot of social life, all that kind of stuff. I don't know. That could also be beautiful.

ThePrimeagen

There could be something very, very exciting about that in some sense, especially if you're building your own thing. I could imagine that would be very exciting. If I was Amazon, Jeff Bezos building Amazon, one could imagine that those early years were probably very rough and the amount of hours he probably put in we're very, very rough. But I will say that there's this unique aspect in our culture where we make this as an equal trade-off between family or work, like "Oh, you do or you don't have to have kids." And my only real notion with that one is that you will never know your capacity for love until you have kids. You just don't know. And some people are like, "Oh yeah, but I love my dog." It's just like I loved my dogs too. And then I had kids and now my dogs are, "They're all right. I like them."

Lex Fridman

I get it.

ThePrimeagen

I could come home and I pet Indy and I'm like, "Oh, Indy." And then I'm just like, "Okay, bye Indy." I can't even describe the difference between the two, it's not even the same. And so that trade-off making is no one can tell you what it's like because there's a real reality that's right now, and I'm sure, I'm 100% positive this is with my wife as well, where if right now we got news that said you have some medical procedure where if we do this, you will die, but your kid will live, there's not a question in my soul that I wouldn't do that. If I could look into the future and if I had to die right now knowing that my kids would've a better life, they would be happier, they'd be more fulfilled and all those things, I guarantee you either my wife or I would take that every single time. It's just like you'll never be able to say that about most things. People will jokingly say that until it's actually on the line. But it's like with that, you just have this ferociousness. I can break out and sweat thinking about somebody fictionally pushing my kid to the ground, actually get real adrenal responses flowing through my body. So it's just such a different world and it's hard to explain. And you could never have convinced me when I was young that it'd be this big.

Lex Fridman

Yeah, yeah.

ThePrimeagen

Yeah. I thought I knew. I didn't know.

Lex Fridman

But to add on top of that, some of the most successful people I know, some of the most productive people I know have kids. So I don't know if it's even a trade-off. That love you feel, it seems to be a catalyst to make sure you have less time, but you're going to use that time better to be productive.

ThePrimeagen

I would argue that it definitely changed a lot of my life and how I approach problems and everything, in a very different way.

Lex Fridman

Let me ask some random questions from Reddit. On a scale of one to 10, how much do you hate every product Microsoft has ever created and why is it a 10? I think we covered that.

ThePrimeagen

We haven't technically covered it.

Lex Fridman

There you go. All right, go ahead. Go ahead.

ThePrimeagen

Okay.

Lex Fridman

Use your time.

ThePrimeagen

The only thing I'll say is that I don't like that Microsoft pretends to be the good guy when what they really wanted to get you addicted to their products, to get you to use their products as much as possible so they can extract as much money out of you.

Lex Fridman

Well, in this world, are there really good guys?

ThePrimeagen

That's a great point. I would argue Neovim is a great guy. There's no way they can make money. Justin Keyes is the benevolent dictator and he thinks deeply about the product and tries to make it the best as possible. Whereas something like Microsoft, they made VS Code as a loss leader. Copilot's probably operating on a loss leader. These things are all getting you so tied into, GitHub, remote workspaces, CI, Copilot, you become this trapped in permanent person and if that price rises, the switching cost is so great at some point that you'll never be able to switch. That's my only fear is that Microsoft was once accused of EEE and it feels like they're EEEing again.

Lex Fridman

Yeah, I'm nervous about criticizing a good thing because you could see an incentive to do that good thing, like Google creating all these services that don't make money like Gmail for example, you can cynically say they're only doing that to tie you into an ecosystem so they can basically keep you for life. But also it's awesome that they created Gmail-

ThePrimeagen

Yeah.

Lex Fridman

... and they created an incredible product, so-

ThePrimeagen

I can side with you on that one. It is a good product. VS Code is a good product.

Lex Fridman

Yeah.

ThePrimeagen

Now, I think don't put that on the... But it is fine. They did a great job.

Lex Fridman

Yeah. So there is going to be financial incentives behind some of these companies. And by the way, me defending, not defending, but saying positive things about Microsoft is just so I could talk shit to Prime. But that's...

ThePrimeagen

I love that by the way.

Lex Fridman

Yeah, Linux is my first and last love, it definitely... The spirit of Linux and open source is a beautiful thing so I do think that when you have these large corporations, even when they try to do good, oftentimes the profit imperative just takes over and they can corrupt themselves and Microsoft has a long history of doing just that to themselves.

ThePrimeagen

Yeah.

Lex Fridman

That said, they've done, they have you could say for cynical reason because they want to seem like the good guy amongst developers, but they've done a lot to support open source. It's just like, same with Meta, Meta has done insane amount-

ThePrimeagen

Yeah.

Lex Fridman

... to support open source. You can say, actually for that one, I don't know if I can even make a financial or a cynical case for why Meta is open sourcing Llama and these-

ThePrimeagen

Yeah, that one's confusing. It just seems great.

Lex Fridman

Maybe for hiring. But no, I think that's legit, just an ethical, really powerful decision. And sometimes these companies, because they have a lot of cash, can make the right, do the right thing.

ThePrimeagen

Yeah. It's a really positive way to look at it and I think that's really nice.

Lex Fridman

Well, we should always be skeptical.

ThePrimeagen

Yeah, I mean because at the end of the day, companies, they're not good, they're not bad, they're morally neutral. It's the people that are running them, the decisions those people make that are really where the bad or the good comes from.

Lex Fridman

Another question, ask him if he knows how to milk a cow? I've already asked that. The answer is-

ThePrimeagen

No.

Lex Fridman

Oh, no, you don't know.

ThePrimeagen

I've never milked a cow.

Lex Fridman

Never milked a cow.

ThePrimeagen

Almost been killed by a cow, but never milked a cow.

Lex Fridman

Did you ever ride a bull?

ThePrimeagen

No.

Lex Fridman

All right. Why male models?

ThePrimeagen

Okay, so I can explain that one. I will say something like, "I really dislike the color purple because the color purple makes me upset." I don't know, just something very benign. But then someone right afterwards will be like, "But why don't you like the color purple?" And it'll just be like... It's just like Derek Zoolander. It's just like I get done on a five-minute talk about

it and then the next question's like, "But seriously why though?" It was just like, "Why male models?"

Lex Fridman

Yeah. So that's the Zoolander reference when there's a long explanation why male models and he agrees and then forgets.

ThePrimeagen

Yep.

Lex Fridman

What is Ligma?

ThePrimeagen

I've died by Ligma quite a few times. So do you know the origin story of Ligma?

Lex Fridman

No.

ThePrimeagen

So Ninja, famous streamer, someone got him with Ligma and said like, "Oh," something like, "Have you heard about Ligma?" And he was like, "No." And he's like, "Oh, Ligma balls." And then after that Ninja got so hurt by getting had by that, he started banning anyone in chat who's said the word Ligma or something like that. And so then if you don't embrace the meme you get destroyed. So of course, gets destroyed and so then the whole goal is that, can people get me with Ligma? TJ did iladies. He's like, "Oh, did you hear that E-girls got renamed to iladies?" And I just didn't even see it coming. And I was just like, "What?" And he's like, "iladies nuts on your face." And then it's just like, "Oh my gosh." And then a pirate software has also got me like, "Oh, have you heard about Google SIMA," which SEMA is a real product by Google> And I'm like, "Oh yeah, I've heard about this. What is this again?" He's like, "SIMA balls," right? It's just like, "Dang it," how do I keep...? So I've just had it happen live on stream many, many times. I've died by Ligma the most.

Lex Fridman

Please ask him about the size of his dict.

ThePrimeagen

Okay, so that's dict, that's dictionary in Python.

Lex Fridman

Who doesn't love dicts?

ThePrimeagen

Yeah, that's a great question. Just a dicts party when you use Python.

Lex Fridman

I love dicts.

ThePrimeagen

That should be a T-shirt. That's actually a hilarious T-shirt. So on Stack Overflow, you can ask any question you want, and I decided to craft a question one day on Stack Overflow that says how to measure your dict and bytes. And then I proceeded to really go to town and explain all the different things like, "Well, what about the cost of the strings and the references?" And when you really get both hands on your dict and really go after it, it's very hard to, really threw in some innuendos. The Stack Overflow team deleted the question, and then someone hand wrote me an email explaining why they deleted the question and complimented me on how thoroughly and thoughtful the question was just to weave in innuendos and that the entire team was impressed, but it's inappropriate and it had to be deleted and don't do it again or we're going to ban your account. And so it was a very funny moment and so I was like, "Oh, that's funny that happened." That was about six years ago. Last year I was at a conference and there's a guy wearing a Stack Overflow name tag and I was like, "Oh, you work at Stack Overflow?" He's like, "Oh, yeah, I do." I'm like, "Do I got a story for you." And he goes-

Lex Fridman

Oh, no.

ThePrimeagen

... "No, wait a second. Are you the dict guy?"

Lex Fridman

Yeah.

ThePrimeagen

That was his only question was that. And I was just like, "Let's go." I didn't even say anything about me and he already knew immediately I was the dict guy.

Lex Fridman

I should say in all seriousness, I think I've had a bunch of conversations in the Python world where I would have to mention the name of this data structure and it makes me uncomfortable every time.

ThePrimeagen

It's a very unfortunate shortening of a word.

Lex Fridman

Dict. It's just like when I go to the hardware store and ask for caulk and there's always a nice old lady and I ask her where to find, and it's very uncomfortable. I try to pronounce it as hard as I can.

ThePrimeagen

Really get that L in there, like caulk.

Lex Fridman

Caulk, just to be clear. And try to avoid eye contact the whole time. You said that God was a big part, was a big part of your life. Can you speak to that a little bit more? Who is God and what effect, what role did he play in your life?

ThePrimeagen

So I did talk about that one important evening where I, for whatever reason, gained my conscious that moment. So obviously for me that I grew up with a life where I would probably argue myself as a functional atheist. I went to church a handful of times. I can't quite really remember actually going to church as a family in any sort of sense. So there wasn't some super strong tie or anything like that to it. Pretty much anyone else growing up in America in the '90s, you had some sort of impact or intersection with church at some point in your life, that was just a very normal thing I would probably say. And so when that happened, it was a fairly big surprise for me. I wasn't necessarily going that direction or deciding to do any of those things. And so for me, it's obviously the turning point of my entire life. I cannot speak to who I would be now without that. I can just tell you that I wouldn't have had the drive. I probably would not have completed college. I would've not have found my wife or had my kids. I wouldn't know how to value people. I don't think without that whole thing, my value for people would've been very, very small because I would've continued to just objectifying in the way I was. And then probably the biggest thing is there's this one verse, I don't even know where it's at, it effectively says that we love because he first loved us. And so for me it's like I don't think I would've ever lived a life that was happy without this. And I just didn't even know that that was an option for me. And I never really, it was a very tough set of years for me and I was very, very sad and just always just constantly looking for something to fulfill me. And so it's like I didn't have any confidence, I didn't have any joy. I felt very sad. And so that was this moment where for the first time ever didn't, all of a sudden I just felt like I didn't have to live up to a standard. The standards have already been paid for, everything's already, that's the free gift, that's the exchange. And so it's just like for the first time, I didn't have to be the cool guy, I didn't have to have all the right words, I didn't have to feel, I didn't have to go on the conquest, the sexual conquest to find validation. I didn't have to do any of those things and it was exceptionally liberating. And so who is God? That's more of a catechism question perhaps. What is man, who is God? Those are much harder questions. I believe that anytime you try to get too deep into describing who God is, you typically fall into Christian heresy. But for you -

Lex Fridman

He gave you a chance to be happy.

ThePrimeagen

Yeah. He gave me a chance not just to be happy, but also made it so that the first time I can actually feel forgiven I guess in some sense, and able to forgive people that hurt me. For a long time I had this weight I'd carry around from the things I hated about high school and all that kind of stuff. And through that experience, I just wrote down every last person's name and actually held them with me for quite some time and this was the list of people I forgave. And I read it a few times. I couldn't let myself be angry or consumed by that kind of stuff because hate is so sticky, it sticks for a lifetime. And there really is only one cure for hate, which is forgiveness. I just don't think you can get rid of it without that. And so I just had choose to forgive these people and to move on, and it really kind of freed me. And I would never have thought forgiveness as a means for that change if I didn't first experience it myself.

Lex Fridman

What's the role of love in the human condition, to go to the philosophical, and what's been the role of love in your life?

ThePrimeagen

It's very obvious that every person wants or desires love. My wife has recently convinced me to watch Love is Blind with her one time. And you watch the show and if you're not familiar with it, it feels like just a disaster of an experiment to just cause crazy filming. But anyways, the idea is that if you just don't see somebody, you can fall in love with somebody and want to marry them after 10 days or some very small period of time. And what you really end up seeing is all these people who are just desperate for actually love. And there's some part of it... I told my wife, "It's like love gladiators." We're watching people battle it out for drama and really what they want is love. And it's like they're fighting to the death and love, if you will. And it's this almost kind of sad aspect to watch. And so I think that it's hard to call, what is its role in the human experience, because I think it's just something that we all naturally not just want, but need. And I don't think that you can really progress, and when I say the word love, I would like to kind of narrow it down maybe a bit more. And I don't mean Eros, the Greek word like sexy love, I think that paternal and friendship love are extremely important. And I think agape, God love is also very important. Agape love is the one that is superior to them all, but obviously different and also co-needed with the parental ones and all that. And so you kind of need this mixture of them all, and each one is different for each reason and where it's applied. And so I don't think... I just don't see a world in which is good of any kind without that as a very foundational piece. Because again, I didn't come here trying to quote any sort of scripture, but it says that it's not the nails that hung on there, it's love. That's the

reason why these things happen. And so if forgiveness is the requirement to kind of pay off hate in some sense, then love has to be the motivation for forgiveness.

Lex Fridman

Yeah, that's the tragic aspect of life. I think there's a deep loneliness in all of us and a longing to be a part of this bigger thing. And that longing is a love and it has many names, but yeah. Yeah, the love aspect of it, it's the beautiful aspect of life, the tragedies, the loneliness, and the unfortunate suffering that is a fundamental part of life and the beautiful aspect is the love.

ThePrimeagen

Yeah.

Lex Fridman

Which I think is a good time to mention more Reddit, the place for everlasting positivity and love. Somebody wrote, "Please thank him, you, for his everlasting positivity and give him a big hug for me." So I won't give you a big hug on camera because I'm afraid I'll get a boner and that'll be very unfortunate.

ThePrimeagen

Hey, let's not bring dicts into this again, okay?

Lex Fridman

It's my favorite data structure. Like I said, I love dicts, all kinds of dicts, ordered dicts-

ThePrimeagen

Unordered.

Lex Fridman

... unordered dicts. I don't discriminate. Yeah, but just that to say big thank you from me. I listen to you a lot and I just really enjoy... I've been going through a lot of shit myself and just the positivity, even when you're building the stupidest shit, it's just the positivity radiates from you and you inspire me to be a good person. You inspire me to build stuff. So thank you. And I'm sure there's many, many others who listen to you for the same reason. So thank you for your positivity. Thank you for being the light in many people's lives, and thank you for talking to me, brother.

ThePrimeagen

Dang. That was very, very kind. I really do appreciate all those extremely nice words even from Reddit. That's very surprising. But thank you. I mean, I know you know that there's many people's lives, and I'm sure you've received the letters that have been changed from actions and things you've said and things you've done. And so it's one of the best parts about

doing this side is that you get a chance to potentially improve somebody's life. And you getting to interview a lot of people, there's a lot of people that listened to Chris Latner and saw his excitement for Swift and probably went and learned Swift and then got really amazing jobs and it can be all originated back to back to you in that interview. And so those are amazing things. And so same goes back to you, you've done a lot of good stuff.

Lex Fridman

Right back at you brother. Thank you for talking today. Thanks for listening to this conversation with Michael Paulson, aka ThePrimeagen. To support this podcast, please check out our sponsors in the description. And now let me leave you with some words from Paulo Coelho. "When we strive to become better than we are, everything around us becomes better too." Thank you for listening and hope to see you next time.