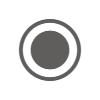
**Transcript**

24 June 2025, 06:59am

 **Steven Joubert** started transcription

 **Steven Joubert** 0:04  
OK. How do you stay current with new programming languages, frameworks and tools?

 **Shaun Richardson** 0:12  
So this is actually this is something I've thought about before, like I don't stay current with frameworks and languages and tools. Specifically, I stay current with people.  
So when I'm going on like this, say social media or I come across people. And if I if they're in the industry and I see them a lot and I like this stuff, I spend, I spend some time with them online to kind of, you know, like a good example be Scott Hanselman. I guess I was like C#. I want to. I want to stay up to date with C# and Microsoft ecosystem. Who do you know? And C# is a developer advocate for Microsoft.  
And he posts, you know, fun and thoughtful and insightful stuff he's got.  
Loads of his own kind of videos and blogs and all sorts. So I follow him.  
Then it really cuts down my work.  
By an astronomical amount and I make sure that the people I follow online.  
I'm constantly kind of pruning them if if what they're if, if what they're providing isn't aligning with my interests anymore.  
Yeah. And and I find that cut, that takes a lot of work out of the equation.  
And then next thing they'll tell you because they're excited like.  
I follow you know Kravitz, she's a developer advocate at Google. And she said, oh, yeah, select you can now do a custom select menu in HTML. And I was like, whoa, yeah, I have no idea. That was close to being delivered. And it just kind of popped up because she mentioned it. So, yeah, that's that's how I stay. That's how I stay up to date on.

 **Steven Joubert** 2:00  
And like you said, it cuts out a lot of the hard work for you and they're bringing it too.

 **Shaun Richardson** 2:06  
So much so much, because otherwise you're following ten different sites then.

 **Steven Joubert** 2:08  
OK.  
Yeah.

 **Shaun Richardson** 2:12  
When all you know.  
You follow people.

 **Steven Joubert** 2:16  
Come on. OK. Next question.  
What piece of technology, past or present, do you feel the most emotionally connected to?  
Noah.

 **Shaun Richardson** 2:30  
It's a good question, man. You know what?  
The first thing that comes to mind is is.  
Lender Lambda calculus really.

 **Steven Joubert** 2:41  
Wow.

 **Shaun Richardson** 2:42  
Is.  
But because for a reason and I'm not talking about lender calculus, if you do like at school, I'm talking about, you know, combinators, so.  
Basically.  
Just to kind of dumb it down.  
There was a way that you could programmatically just take two simple axioms. One was like two and one was false, and just by playing around with them you could basically create a fully Turing complete computer programme, like a language.  
And I'll send a link for you the HJKHKO combinators.  
And the reason that resonates with me is just a combinatory logic. The reason it resonates with me is because.  
A while ago, something insanely smart people took two of the simplest things that you can explain to a baby and made a complete programming language out of it just by building it up and composing different parts together. And like, let's just.  
Ticks all my boxes, you know.

 **Steven Joubert** 3:58  
Like a. So it's essentially the. Essentially it's the the the Lambda calculus is the basis of programming, like at its fundamental that you can then put into. I think that was when you went down the pro, the object orientated. Or is it the function the functional language programming, yes.

 **Shaun Richardson** 3:58  
Efficiency.  
Yeah.  
Functional, functional, prone. Yeah. And and OK. Like, just to be honest, it's it's not practical in any kind of scenes. But it was the purity of the idea that the people they just went way further than anybody.  
Sane would run with it and they just carried on and and fundamentally proved like, hey man, we can really go quite far with very little and I think that's what appealed to me is that they have these two very simple, elegant axioms.  
That they could just adapt and compose and extend. And that's means like the real heart of programming and and even though it isn't really a technology 'cause, it's easy to say like AI or whatever. I think, you know those kind of discoveries that happen in the past, they've just laid the foundation for everything we're feeling right now, you know.

 **Steven Joubert** 5:07  
Very cool.

 **Shaun Richardson** 5:08  
Yeah.

 **Steven Joubert** 5:10  
OK. Is there a tech train, do you think is overhyped and one that is not getting enough attention?

 **Shaun Richardson** 5:20  
Do you think most tick trains are overhyped?  
Because I think there's a lot of people who want to make money and are in a position to make money in the beginning.  
When when understanding the store isn't, you know there's like the the the initial swell of hype, especially like every time a new technology comes around like Bitcoin or something, there was a huge swell of hype and and eventually it kind of settled kind of settles in the middle.  
On something new comes out. It was the cloud not so long ago, the cloud was like, oh, everywhere. And then, you know, it was rendering websites on the edge. And, you know, and now it's AI. And I'm not saying that these things aren't amazing technologies. They are. I do think there's a period in the beginning where almost all of them are over hyped they eventually find.  
Like they settle, you know? And will they change all? Absolutely.

 **Steven Joubert** 6:21  
Yeah.

 **Shaun Richardson** 6:22  
You know, but.  
That's it's always fun to think about that for things that are under under heart. I think there's so many things that we don't even know about.  
Where you know AI and and Bitcoin and all these things kind of came out of the blue and they just took over the world.  
And then.  
You know, we we all know about them, but I think for all of those, these are technology we didn't know about. There was some development for 20 years.  
And and when it got released.

 **Steven Joubert** 6:58  
OK. Yeah. We we're just in the face of everything. Meanwhile, they're underlying progressions of of, of, of, of programmes that have moved along.

 **Shaun Richardson** 7:05  
Yeah.  
Exactly. You know, it's just like GPS is a great example. GPS was like a military funded technology and it was developed for many, many years to make the army more efficient.  
And then, when the scientists developed the technology, someone said, hey, this has, you know, consumer application. And when the military and the government said yes, let's, let's open up this GPS to kind of.  
Consumer all of a sudden there was booms in GPS, garments were put in all the cars, phones started getting GPS S and it's just that technology unlocked the whole new ocean.  
Of of benefits, so like to answer your question, I think they are at hand. They're under height and I think you know they're like microwaves. When they eventually come around or like, of course, of course they're here.  
But they've been developed for many years before then.

 **Steven Joubert** 8:15  
There must answer.  
What piece of or piece tools or pieces of software you can't live without in your workflow?

 **Shaun Richardson** 8:29  
I'm not even to a VS code nowadays. You know, I grew up, grew up and I started with my career doing them. I love my editor, but I kind of have grown out of it almost. It's just it's too much effort to set your editor to all these different language and environments that you might be using on a database.  
Vs code is just great, just fired up, it just works.  
So I always have there close by. I always have a scratch paired like some sort of notepad that I can sort ideas down and then throw them away.  
And always have calendar.

 **Steven Joubert** 9:08  
Or physical.

 **Shaun Richardson** 9:10  
For the both actually I have notepad, the actual programme on my computer right now and it's got it's got about 20 tabs open.  
I'll just. I'll just go new tab and I.

 **Steven Joubert** 9:22  
I can. I'm looking at the exact same screen.

 **Shaun Richardson** 9:26  
Yeah. Yeah, it is 20 times or just like every time my wife says, oh, don't forget milk. And I'll open an ETF and say milk and and. But I also have a physical pair because sometimes I feel like I can't think on it's drawing and scratching and crossing out. And so, you know, like when it comes to tools, that's very minimal. It's my editor, a scratch pad of some sort and a calendar.

 **Steven Joubert** 9:47  
Yeah.  
Use any sort of plug insurance for your VS code that will that will help you with generating code.

 **Shaun Richardson** 10:04  
So there is.  
You know.  
Copilots is installed.  
And it's OK.

 **Steven Joubert** 10:15  
Do you do? Do you do you lean on it? Do you go? Is it a go to or is it just an ask to have that? You found that that you is it. Is it part of your tool kit?

 **Shaun Richardson** 10:24  
I don't.  
No copilot would not be as much Potomac toolkit as like I'll just open up a chat TPT tab and say I want.  
Like a class that does XYZ and it put prints out a stub for me and I might, I might reprompt it to get it sort of into the shape. And the only reason I do that is to type less really.

 **Steven Joubert** 10:46  
OK.

 **Shaun Richardson** 10:58  
I find that.  
The copilot doesn't give me.  
Any sort of code that I couldn't have got to myself.

 **Steven Joubert** 11:08  
OK.

 **Shaun Richardson** 11:08  
So I use it more as a workhorse, like a donkey.

 **Steven Joubert** 11:12  
OK.

 **Shaun Richardson** 11:13  
Like a trick I do do is if I have to do something repetitive, I'll just like I need to do let's say 10 different test cases for this one thing I'll just drop down 10 lines of the smallest amount of code I can time and then I'll tell GPT to flesh it out into full blown tests, you know.

 **Steven Joubert** 11:32  
Oh, OK.

 **Shaun Richardson** 11:33  
You know, I use it more like a a a workhorse. I feel like every time I try to prompt.  
Copilot to be clever.  
I just felt that it didn't really deliver. I mean, it's probably going to change, you know, but right now.

 **Steven Joubert** 11:50  
The job's still same.

 **Shaun Richardson** 11:53  
Maybe you know, but I found it formal, usable as like a rabbit than anything else.

 **Steven Joubert** 12:04  
How do you balance experimental with practical execution in your projects?

 **Shaun Richardson** 12:12  
Her balance, experimental and practical. So like I'm always trying to experiment because.  
My my default mode is practical and if my default mode starts becoming abrasive then I realise that there's room for improvement, but very often I'm I'm coding on a deadline. Very often people aren't paying me to be creative or solve a problem outright, so I usually try tack it in the corners.

 **Steven Joubert** 12:33  
No.

 **Shaun Richardson** 12:47  
If you will, you know. So I'll do the pragmatic approach and then I'll maybe spend like let's say 5 minutes or something, a little creative to see if it's if there's any light at the end of the tunnel for this approach.  
You know, I don't spend huge amounts of time being creative. I just try and get it out.  
Because usually when when you know your problems wrong, like it's very apparent that it's wrong because things are breaking, it doesn't make sense. Other people to do your code and they say oh.  
It's terrible. It's very apparent, you know, in the beginning you don't even know what you're trying to solve, really. You're just trying to get all the ideas down into some code. So there is always experimental lines and functions, and maybe I'll try a little thought in a scratchpad or a ripple or something.

 **Steven Joubert** 13:29  
Yeah.

 **Shaun Richardson** 13:44  
Yeah, that's kind of how our balance. It's like I'd always keep my progressive hats on with maybe a sneaky experiment hat like off to the side.

 **Steven Joubert** 13:56  
OK, very cool.  
Thank you. That's that's all I need. I've got lots of data. Lots of.

 **Shaun Richardson** 14:00  
Record.

 **Steven Joubert** 14:06  
Lots to work with here.

 **Shaun Richardson** 14:08  
Nice. Yes, you could us.

 **Steven Joubert** 14:09  
I want to talk about the Excel dude. The Excel I needed to create an MPV function.  
From a spreadsheet.

 **Shaun Richardson** 14:19  
Yeah.

 **Steven Joubert** 14:20  
The clients seen through their like this is how we calculate our NPV and I'm like a choir doesn't do it that way. They're like, no, we want the answer to be that I'm like, oh, my word. I don't know how to do that. I uploaded the spreadsheet to GPT.

 **Shaun Richardson** 14:31  
No.  
Yeah.

 **Steven Joubert** 14:39  
And I said I need AJ S function that will get me Sol C7's answer.  
And it gave me an MPV calculation to. It gave me an MPV calculation that matches exactly. I'm like my word. Yeah, it it it actually it it did. All the brain stuff for me. I'm like, OK, so I copied and pasted that function and it it actually. I'm like, wow. So I think I think because.

 **Shaun Richardson** 14:49  
And it's it it.  
Wow.

 **Steven Joubert** 15:10  
You don't have to worry the problem, it understands the problem to at because it's taking cell. Yeah, it's taking axe cell because it Excel used an MPV calculation.  
A known excel calculation.  
Programme.

 **Shaun Richardson** 15:28  
OK, a known built in function.

 **Steven Joubert** 15:29  
And because it's an, it's an Excel MPE calculation. Yeah. So because it's known that there was no ambiguity in solving the problem.  
And so it was able to literally spread out a working Excel JS function for me. I'm like man, that was awesome. Yeah.

 **Shaun Richardson** 15:50  
Damn, that's pretty cool.  
Yeah, job done then, eh?

 **Steven Joubert** 15:54  
Yeah. So I think it's a client and the client's like, thank you. I'm like you're welcome.

 **Shaun Richardson** 16:01  
That's cool, babe. Like, I mean, exactly because The thing is like, you'd have to sit through that and sweat through that for some magic number. That's what it is. It's a magic number formula.

 **Steven Joubert** 16:03  
Yeah.  
'Cause, I don't know. I don't know exactly how excel is calculating this because Excel's calculation out of MPV is different to if you Google MPV calculation.

 **Shaun Richardson** 16:15  
Yeah.

 **Steven Joubert** 16:21  
There're there how they're getting there is a standardised thing, but there's differences, there's differences. Why Excel isn't exactly correct to what our our system calculates because their start of day, sometimes they have a zero interest on the first day there, there's differences that are specific and excel.

 **Shaun Richardson** 16:33  
Yeah.

 **Steven Joubert** 16:42  
Is it this way?  
And and and and if you Google it, the web has got several ways to do it, so I want I want the Excel solution and yeah it it just popped out. I'm like, wow. Thank you.

 **Shaun Richardson** 16:50  
Wow.  
You look crazy. Yeah, I mean, like, you know, also, the a lot of that stuff is gonna be very, very valid. Because, I mean, just like all the time that that chews up, what it used to chew up. But I also think that that you, you know, it's like everything else with power, it's very easy to go down this line.

 **Steven Joubert** 16:59  
No.  
Yeah.  
Yes.

 **Shaun Richardson** 17:21  
So Google right? Google has a policy. The further the further the bug is into the system, the more expensive it is to resolve.  
So if the bug is in your editor as you're typing, easy, super easy. You just gotta fix it. Doesn't cost anything, but if it goes on to your manager then your manager takes a bit of time to spot the problem, then it's going to come back to you. You've got to redo it. This and so on and so forth. And it goes into the branch, and then it gets tested in the CR pipeline. And then if it's in production, oh, you have to release.  
A press release saying blah blah blah. Like the further you know it costs.  
Money to resolve it and I think because AI makes it cheap, to go further.  
I think there's going to be more problems further down than our more expensive to resolve because you can effectively have one guy with GPT doing the job of team and team stickers.

 **Steven Joubert** 18:15  
Yeah.  
Yeah.

 **Shaun Richardson** 18:23  
Yeah. It's like we'll make the mistakes and we'll eventually settle on like a accepted method of using it. You know, we're we're kind of in cowboy time now.

 **Steven Joubert** 18:37  
Woman, I want to do you. You said you're following some your VS code. Do you have?

 **Shaun Richardson** 18:40  
Good.

 **Steven Joubert** 18:45  
A guide for setting up your VS code environment.

 **Shaun Richardson** 18:50  
Yeah, I can. I don't have a car, but I can hook you up real easy. Super simple. And I've got some plugins that I use.  
Yeah, I'll be happy to help man.

 **Steven Joubert** 19:01  
I'm not trying to if you could put something together and I can link it.

 **Shaun Richardson** 19:09  
Oh yeah, I'll put it on my blog.

 **Steven Joubert** 19:12  
We go and then let me know and then I'll link it.

 **Shaun Richardson** 19:14  
Yeah.  
Send you. OK, so all all the things I'm how I set up my VS code.  
I'll set up my VS code and then can I ask you for a final favour?

 **Steven Joubert** 19:23  
Yeah.  
Yeah, you wanted that reference.

 **Shaun Richardson** 19:27  
Just one more, just a sentence. Yeah. So I can put it on my front page.

 **Steven Joubert** 19:32  
Oh, OK, cool. OK, cool. I'll do that for you now.

 **Shaun Richardson** 19:36  
I'll check with you. No, no.  
Yeah. Thank you, buddy. Thank you. You're successful morning.

 **Steven Joubert** 19:44  
Do you do you need the trance you've got the transcription from the the last one.

 **Shaun Richardson** 19:50  
Yeah, I've just gotta open it up. But if if I come across problems, I'll let you know for sure. I've also dropped you. I've dropped you a link about the skr Combinatorials bringing up.

 **Steven Joubert** 19:52  
OK, cool. No, that's fine. If you need anything else, give me a shout.  
Yes, thank you. You if you yeah, whatever link you can.

 **Shaun Richardson** 20:08  
That link there.

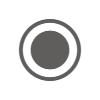
 **Steven Joubert** 20:08  
I have just added to the post. Cool I see that.

 **Shaun Richardson** 20:13  
What's a lot, bro?

 **Steven Joubert** 20:13  
Gross. You asked me and this is cool. Yeah, this was a fun.

 **Shaun Richardson** 20:15  
See you later. It is funny. Yeah. Cool, man. And check on her. She's there, right?

 **Steven Joubert** 20:22  
Yeah, man. Cheers, bud. Bye.

 **Steven Joubert** stopped transcription