

cps-13

📅 Wed, 7/6 7:38AM ⌚ 20:15

SUMMARY KEYWORDS

python, code, micro python, nicholas, music, grade, beginner, bit, circuit, world, editor, people, learn, bbc micro, project, freelance musician, community, education, mu, coder

SPEAKERS

Paul Cutler, Nicholas Tollervey



Paul Cutler 00:03

Welcome to the circuit Python Show. I'm your host Paul Cutler. This episode I'm joined by Nicholas Tollervey. Users of circuit Python might know Nicholas is the creator of the mu code editor. Nicholas is also known for music, philosophy, teaching, writing, and computing. Nicholas, welcome to the show.



Nicholas Tollervey 00:20

Hello there.



Paul Cutler 00:20

I'm glad to have you. You've had a very interesting career. How did it all get started for you?



Nicholas Tollervey 00:26

Back in the 1980s when I was a kid, the UK Government decided that every school should have a microcomputer. And that was the BBC Micro, my mum and dad were both teachers. And my dad, one half term brought home his school's BBC Micro thinking, I'll just figure out how it works and what it what it does, and how can use it in the classroom. And it took about half an hour for my brother and I to sort of price it out of his hands. And for us to get going on it. It's basically started from there. And so all the way through my teenage years, I was using these sorts of computers and things and then I kind of got distracted with music, which I'm sure we'll come back to. And when my own children arrived on the scene, I got back into computing, because it was a lot more lucrative than being a freelance musician, I can tell you, and that eventually led to some work for the BBC, the BBC, as you probably know, created a device called a micro bits where they wanted to recapture those heady times of the 1980s, where, you know, kids were learning to code on BBC Micro computers. And they asked for people to contribute to their project. And I was a PSF Fellow at the time, and they mentioned Python, with the blessing of

the PSF boards. I said, Well, hey, you know, you should talk to the PSF. I'm a PSF fellow we can coordinate. And so I became involved in the BBC Micro bit project, I wrote a lot of the software that's to do with Python, for the BBC Micro bit project, a lot of the documentation is mine, as well, and a whole bunch of other stuff. And I got to meet an Australian gentleman called Damien George, who is of course, the creator of micro, Python. And Damian, I don't know, it took him a weekend, he's got a brain the size of a planet, that chap to basically get micro Python running on this device. And so that's how I got into Python, hacking computers, the circuit, Python and micro Python ecosystem, and it kind of all went from there, really.

P

Paul Cutler 02:34

So you started working on the micro bit, when that first came out six or seven years ago? How did that lead you to the mu code editor?

N

Nicholas Tollervey 02:41

Well, one of the tasks that I had been assigned was to create a browser based editor for micro Python on the micro bit. And being the sort of engineer who likes to check their work, I ran a series of workshops with UK based teachers and developers and I'd bring them all together. And you know, all developers want to play with the micro bit. And all the teachers would want to have a go with it and see what they could do with it in their classroom. And it was a good place for me to try out the tech that I was creating at the time to see does it work. And it became clear that actually, it worked. But that could work better. And things like back then anyway, it's changed. Now back then, if you wanted to flash the device, you had to download the thing, you have to find thing in the file system, you had to copy the thing, paste it into the other thing, wait about 10 seconds, and then see scroll across the screen syntax error line three sort of thing. Okay, so it's a bit of a pain in the neck. And so the other aspect of micro Python is that you get a reple on your devices, you know, and this is great for interactive programming. And when I demoed this with a hacky script that I'd created in Python, teachers were like, Whoa, you can it's like talking to the micro bit directly. Yeah, absolutely. This is what Python can do. And developers enjoyed the fact that there's a reple, because it's an exploratory, playful way of getting into the hardware and seeing what it can do. So putting all that together, I decided one afternoon, one Sunday afternoon, yeah, how hard can this be if I were to write a very simple code editor that had a flash button, instead of all the download, blah, blah, blah, chicory? pokery. You just click flash, and then it just kind of work you. So you reduced the sort of dissonance in the development cycle, if you see what I mean, you were writing code flashing, writing code flashing, or you could drop into a reple and then typing away, and that's the origin story of mu. And for Sunday afternoon hack, I think I'm doing quite well.

P

Paul Cutler 04:38

I think it's turned out great. It's the go to editor for so many people because like you pointed out having access to the reple right there makes it so much easier trying to troubleshoot or change whatever you're trying to work on. Yeah. Yeah.



N

Nicholas Tollervey 04:50

The other aspect of me is because it's a beginners editor, or it's aimed at people who are supporting beginners. We are not trying to do that. Big whiz bang effects of an IDE or something like that, we just want to do the simplest possible thing so that people can graduate away from mu, and then use their VS code or whatever it is they graduate to. But the side effect of that is it just does what it says on the tin, it means you've only got a few buttons across the top. What does it do it does that, go look at those buttons and explore and play. And we've built it in such a way that it has to cope with the working practices of 11 year old beginners. So you know, they are clicking like nobody's business. So we try and make new, quite robust in that sense as well. I'm really pleased that people find it useful. Oh, it's

P

Paul Cutler 05:40

great. You had mentioned when he first tried starting to code mood that you were looking for an online code editor is that idea still in the back of your head,

N

Nicholas Tollervey 05:49

said the online code editor was a part of the BBCs requirements. And that ended up becoming make code. But the folks that make code have taken out the micro Python bit of that and put in their own Python thing now. But I do know that the micro bit Foundation have a new web based browser based Python Editor. Because handily enough, Carlos who works at the microwave foundation is also one of the core mu contributors as well. So I get to chat with him, perhaps every fortnight and see what they're doing. Looking over the fence. I'm very excited to see what they've been doing with that. And I've heard just yesterday from us, electrical engineer friend who's been testing it that he said, it's amazing. So when I see that I'm going to be a happy little shot.

P

Paul Cutler 06:37

You'll move off, it's great to see that there's continuing innovation in that space. Absolutely. You're active in music making, what is your music background?

N

Nicholas Tollervey 06:46

Okay, so the part of the UK where I grew up is, I guess, well, it's called up north, really. And up north, there's lots of coal mines and pits, and all that sort of stuff. And each of those villages has a brass band, and I grew up there was a band in my village, and I'm a tuba player. The way music education works in the UK is once folks realize that you're enthusiastic about music or talented about music, you're kind of put on a pathway of what's called the grading system, where it's a bit like belts in martial arts, really, you know, you start with the white belts and eight or nine belts later, you're a black belt. It's the same sort of thing in the UK and other Commonwealth countries as well. And so I was put in one end of this sausage machine and went round around around this mechanism. And out came Nicholas attending the Royal College of Music. It's one of the world's top music Conservatoire. It's really I wasn't really expecting to get there. But they let me in. And within about six months of arriving, I realized that actually a

professional musical life wasn't what I wanted to do. I wanted to do other things. But I wanted to complete my degree, I met my wife while I was there, which is the best thing that came out of that. And I also worked as a freelance musician for a few years after graduating, and importantly, as a music teacher, which is what all musicians do when they want to have a stable salary. Because clearly being a freelance musician is the worst job in the world if you want to earn regular money. So that's, that's my background. I'm a tuba player.

P

Paul Cutler 08:21

How is learning to code and learning music similar?

N

Nicholas Tollervey 08:24

That's an awesome question. To learn to play, I don't know something like a tuba or a violin to a very advanced level takes many years of practice sustained efforts, self analysis, if you see what I mean, you have to think about what it is that you're doing reflecting on not only how are you holding the instrument, but how are you engaging with music, it means learning from lots of different people. It means putting yourself into lots of interesting situations like playing in an orchestra or playing the string quartet or a brass band, blah, blah, blah. All of these things and the music world knows this. And so there are structures in place like the grading system we mentioned that help you get from scratching away and entertain the local cats to it being Nicola Benedetti, a virtuoso concert violinist. It's okay. Now, the coding world has yet to catch up with this. But I think we could all agree that to become a good coder takes a long time. It requires self reflection. It requires getting experienced by learning from others and put, you know, all those kinds of similar things to the music world apply to becoming a good coder. Sadly, in the coding world, you'll see, you know, learn Python in three months. And, you know, learn Perl in 24 hours or everything you wanted to know about JavaScript, but we're afraid to ask in four easy lessons or blog posts or books to that effect. And that's great. We want people to feel empowered to learn to code, but I also feel a little bit sad that actually what they're being sold is a bit of a false promise. You know, you and I and anybody listening or watching will know Are that coding? Is it No 99% being frustrated at why the damn thing doesn't work and finding out on Stack Overflow, and you know, maybe 10% actually writing code, it takes the acquisition of a certain sort of mindset and skill set that gets you to a place where you can be effective and autonomous as a software developer. And I think those in the coding world could learn a lot from the music world as a result of that, because they're both very similar trajectories.

P

Paul Cutler 10:28

I didn't learn to code until I was older. And I came across this, I wish I could give credit to whoever said it. But they said I'm not self taught, I'm community taught, it really stuck with me, because as someone who was, you know, quote, unquote, self taught how much I've learned from other people. And one of the reasons I started the podcast was how much help I got around circuit Python. Yeah, a year or two ago when I was starting and realized, okay, that kind of a community is so important.



N

Nicholas Tollervey 10:53

And just just a shout out to the folks that Adafruit who have built, helped build this community and guide this community. What they are doing is best in class, they are phenomenal at this. And we could all learn from the way they do their community management, the way they welcome people, the way they organize things, and just the way they embody what it is to be a friendly coder in the community is something to behold I have a list of names that I'm thinking of, and I'm sure you have, and they might even have been guests on your podcast, but so I won't embarrass them by mentioning them now. But you know, awesome work at Adafruit. Great stuff, keep it up.

P

Paul Cutler 11:35

Your current passion project is called Code grades directly based upon your music education. Tell me a little bit about your project.

N

Nicholas Tollervey 11:42

Okay, so I spent high has spent since 2012, an awful lot of time thinking and working in coding education, actually 2009 That's when I set up the Python code Dojo where beginner coders got together, I set up the education track at PyCon. UK as well and did a whole bunch of other stuff. It was actually going to meet Dan, in a circuit Python, in Boston on a train journey that I had the idea of, well, what would music education that like if we taught it, like we teach coding, you know, learn the tuba in 24 hours, you know, everything he wants to know about the violin, but are afraid to ask him three months or four easy lessons, you know, that's not going to fly? Would you hire a musician to play at your wedding? If they've just been to a three month boot camp? That sort of question. Okay, so that led me to ask, Well, how could I flip that table? flip, flip the tables and say, What would coding education that like, if we did it, like we do music, and I realized, you know, I'm a classically trained musician, I could use the stuff that I learned both as a student and as somebody who had to teach this sort of stuff. And bear in mind, we've got 1000 years of Western musical history to draw upon. So that part of me was being that lazy engineer going, Well, surely, somebody's done this before that I can borrow the ideas, and so on and so forth. The other aspect of this is that my wife, still a music teacher, and she's a very good music teacher. And I see what she does all the time. And she sort of inspires me to think well, how would that work in the world of code. So we get code grades, which is just one aspect of music education here in the UK, which is that kind of pathway like martial arts belts, you start at grade one or white belt, you present a very simple project that demonstrates certain core concepts that are quite simple, because it's grade one, you present that to a mentor, and in a process that's a little bit like code review, they assess you, they get an idea of your level of attainment, and then you get a written feedback and score at the end. So you've passed your grade one with a mark of 75, which is passed with marriage, you go up, that's good. And at that point, the requirements for grade to come on your radar and feel within reach. So it's like stepping stones, you know, if you were to look at grade eight, or black belts, and you're from the perspective of a white belt, or a grade one, it would look impossible, but by going through all this process of the stepping stones, you're moving towards gradually place where you feel that you can do those things. And the other aspect of this, which I have personal experience of is that it's it's a great antidote to impostor syndrome. And so when I was a young musician, one of my music teachers said, You should audition for the local youth orchestra. And I was like, Oh my gosh, I can't do that. They're so amazing. And they're also all big kids, and I'm only 14 years

old, blah, blah, blah, blah. But the minimum requirement necklace is that your grade five and you've just passed your grade six, so you're definitely qualified enough to do this. You can't argue with that. And actually, it made me feel well, okay, maybe I am good enough. There's a lot of imposter syndrome in our world of coding and by Having an independent third party or code mentor, who's a professional developer, assess your project for a particular grade demonstrates that you know what? Yes, you are. But grade five Python level, you know, don't beat yourself down, you deserve that. And you deserve that mark at 75, or whatever it is a high high ish mark. So you know, it's a way of helping people get the wind in their sails interact, you know, learning from the community, as you say, by interacting with professional software developers who are going to be the kind of mentors assessors, and it's a way of finding the smallest steps that get you across the big step. That is the chasm between, you know, a complete beginner to somebody who's a competent coder. That's great. In a nutshell,

P

Paul Cutler 15:49

we've been talking a lot about education, you've taught music when you were younger, to the work that you've mentioned earlier, at PyCon, to the microbead, to code grades, was that a purposeful decision, a conscious decision, or just coincidence on your part?

N

Nicholas Tollervey 16:04

5050. I'm a very intuitive sort of person. So I follow my nose. And perhaps the places where I go, oriented me more towards perhaps educational things, or, you know, show that that's the case. But I have this theory, when I want to learn something, actually, I try and teach it as well, because in order to be able to teach the thing, you have to have internalized what it is that you're trying to teach, and you have to understand it well enough that you know how to make analogies or use metaphors. Or given that this is a particular sort of beginner, I'm not going to blind them with all this science, I know which of the simple concepts to introduce them to without overburdening them or what order to introduce those concepts. And that clarity of thinking is a really great way of learning yourself. So teaching is a great way of learning. And I often say to folks, you know, well, let me try and explain that back to you. And tell me when I'm an idiot when I've got this wrong. And that just works for me, I guess. Yeah, that's, that's part of me, really.

P

Paul Cutler 17:10

We're so grateful that you've been involved with education for all these years, all the things that you've brought to the table have just been great.

N

Nicholas Tollervey 17:16

Thank you. Thank you.

P

Paul Cutler 17:18

It's been asking all the questions. And before we run out of time, I wanted to give you an

I've been asking all the questions. And before we run out of time, I wanted to give you an opportunity to ask a question.

N

Nicholas Tollervey 17:24

Why yes, I do. So you're an experienced open source collaborator and sort of member of the circuit Python community. So I don't know an educational benefit. So what piece of advice that you didn't know you needed to know? Would you give to your beginner self? Okay, so this feels very Donald Rumsfeld kind of there are known knowns, there are known unknowns, what's your kind of unknown, unknown that you now know?

P

Paul Cutler 17:52

And you've touched on it, and it's kind of simple, it's just do it jump in the deep end, put that impostor syndrome aside, and it sounds so much easier to do than it is I understand that. But I was lucky to have a mentor and open source when I first started in the good old community. And that was basically her advice was just go do it. This is open source, no one's going to turn you away. The very rare project that's toxic and isn't open to newcomers who want to help. The best advice I can give you is reach out to someone, tell them you want to help or submit that pull request, submit that patch, write some documentation help with project management marketing, there's so much more to coding and open source projects than just coding as well. That would be something I would tell myself because I didn't know how to code when I first got involved, which didn't help with the imposter syndrome either, because everyone else around me could code but honestly, that's that's the one thing I wish I knew then, that I hadn't waited as long as I did to get involved in. Those are some lifelong friendships I've made in open source community, and it means so much to me. And this is just, you know, one small way of giving back via the podcast,

N

Nicholas Tollervey 18:58

I was just gonna say and here we are. Exactly.

P

Paul Cutler 19:03

Nicholas, last question for you. You're going to start a new project, which microcontroller Are you going to reach for circuit playground

N

Nicholas Tollervey 19:09

Express. It's kind of the mother ship is it was the first circuit Python board that I became familiar with. It's the one I used in the micro Python book that I wrote for O'Reilly, Lemar and her colleagues, Adafruit have put so much cool funky stuff on something that's so small, I wanted to do an example project on code graze the thing I thought of was using one of these

with a COVID mask to do a kind of an emoji type thing. You know, you could make the with the LEDs. NeoPixels make it look like it's smiling or it's it's frowning. You know? It has to be this is so fun. It's so easy, and it's so cool.

P

Paul Cutler 19:48

I couldn't agree more. It was the first circuit Python device I bought and it was a gateway. It just opened up all these doors for me. Nicholas, thanks so much for being on the show.

N

Nicholas Tollervey 19:56

You're welcome. Thank you for having me.

P

Paul Cutler 20:00

Thank you for listening to the circuit Python show for shownotes transcripts and to support the show visit circuitpythonshow.com Until next episode stay positive