

Jess Ingrassellino

LXF: Jess, thanks for your time today. Please can you tell the *Linux Format* readers a little about yourself?

Jess Ingrassellino: Hi everyone, I'm Jess and I'm a software testing engineer by day and at the weekends I teach Python to kids aged eight to 18. I've also been working on curricula and methods to reach out to the kids and to help teach teachers who are interested in technology, but don't necessarily have the access to how to implement technology in the classroom.

LXF: So what brings you to Cleveland and PyCon this year?

Ji: I'm at PyCon this year because I'm the chair of the education summit, which I've been for the past two years. I started volunteering in the education summit five years ago because I wanted to see kids, like my students, who come from inner-city schools.

I wanted them to have a way to know about technology. Children are so often consumers with their phones and tablets. I think that the power balance is shifted when they can also become creators of that which they consume. So back in 2012 I hadn't written much Python, mostly just for work, but when the call for volunteers to organise the summit I thought, "Hey I'm a volunteer, I'll do this!"

LXF: The education summit is where educators meet to discuss how they teach Python across the world.

Ji: There's a lot of value in coming together as an international community and talking about the challenges that we each face in our "little corners", and it's difficult to reach out and talk to people throughout most of the year. Not because we can't (the internet's got that covered), but in terms of finding the time and having discussions.

We also reach out for what we need, and may not be receptive to what others are trying to tell us if it doesn't immediately fit into what we think we need. It was kind of fun to bring other people in who might be saying something a little outside of the box, but it might be something that we might learn from or provide inspiration. I think that it's really important to bring teachers, programmers and innovators together in these spaces, with the opportunity to share and be inspired.

LXF: There are many way for children to learn Python, and one popular way is

via online courses and portals such as [repl.it](#) and [trinket.io](#). Are they useful as education platforms?

Ji: I discovered in my teaching that the web route has been helpful in getting children started with something quickly, but what I choose to do depends on the students.

My current curricula is all online via [repl.it](#) and we use another platform called [py3.codeskulptor.org](#) which is very similar to PyGame, but it's fully web based. By using these tools we can create basic 2D games, for example *Pong* and *Plants vs Zombies*. I use [repl.it](#) as a code platform and a storage area because codeskulptor code only runs on the website, so you can't run it with [repl.it](#).

I've given some of my students them the micro:bit as an experiment. I've noticed that different learners take to different things and I think that for me, that makes the decision. Systemic operation – the traditional classroom approach – isn't how learning really works anyway: "If the child understands then do the thing, but if they don't then fail them." This isn't realistic and the purposes of education need to be revisited.

LXF: It's interesting that you favour tailoring the learning to the needs of

the child. Typically, learning is session based with children learning a set amount per term.

Ji: This is similar in schools in the US. One of my favourite things about teaching tech here is that it hasn't been defined as to what it should look like. The US is very fragmented and that comes from the fact that education is considered to be the state's rights and responsibility. It's not considered a federal responsibility.

We have the common core curriculum, which is voluntary and people have elected to follow it, but it's not required. Technology



Jess is the driving force behind the PyCon Education Summit. Educators travel from across the globe to share ideas and best practices in how they teach Python to their students.

is just something that so many people don't understand how to implement, and it's something that I've seen go wrong in many schools. They don't know what they want kids to get out of it. All people seem to see is that "kids can make money by getting jobs involving code, so we should teach it!" But this is what makes it fun, because I can experiment.

CODING WITH A SENSE OF RHYTHM

"I've boiled all of my tech teaching down to a jazz-improv style. Jazz musicians know their basics, but don't know what they're going to do on stage. I want to provide that framework"

I've boiled all of my tech teaching down to a jazz improv style. Jazz musicians know their basics, the tunes, the melodies, chords and so on. They don't know what they're going to do on stage, but they know the pattern. I want to provide that framework, so I think of my lesson plans not as a script, but more of a progression using the core concepts and "these are the things you need to be able to do, these are the skills that you need to have, once you have your skills (music scales) and once you have your core concepts (chord progression) you can build whatever you want because this is for you!"

