Interview with John Lees-Miller

Paulo Ney de Souza

This interview took place on 24 July 2022, during the TUG 2022 online conference. John Lees-Miller is co-founder and CTO of Overleaf.

Jérémy Just (JJ): Interviews are always highly awaited in the TUG conferences. So now I'm pleased to leave the chair to Paulo Ney for an interview with John Lees-Miller, the co-founder of Overleaf.

John, Paulo, it's up to you.

Paulo Ney de Souza (PN): Thank you.

Welcome, John. The idea is we have a conversation about what you do and what your interests are and so forth.

So the first question I want to ask you is, what was your first contact with computing? Can you tell us how it happened, even if you can remember it?

John Lees-Miller (JLM): It was a pretty long time ago, but the first one I can remember was in primary school, so I must have been like seven or eight years old. We had some old Apple IIs in the library.

I remember they ran a small number of programs. One I can remember was this touch-typing game. We had to race by touch typing, and I was really bad at that. I could not get my head around how people could remember so many buttons. So I didn't get off to a great start with computing, but fortunately my school had a few other computers. And within the next couple of years, I remember there was a Tandy, which was an old make of MSDOS-compatible sort of PC, and it had Quick Basic on it.

Some of my friends and I started learning how to write simple programs on that. I think most of them were like text-based adventure games. So a lot of ifs and elses, that was about it. It was a good system, though. Pretty easy to learn.

PN: And when were you introduced to TeX and LaTeX?

JLM: Pretty late, to be honest. I think I started around 2006 because I was doing an internship, and that's something we'll probably come back to. I was interning at a company that's working on self-driving vehicles and we had some mathematical modeling to do.

And so most people were writing things up in Word and I wanted to try something else. So I think I started with LyX, the L-Y-X, the "what you see is what you get" editor. And I wrote up a few papers in that, and then eventually I realized that I could just write the LaTeX source, and then I switched to it after that. So, yeah, pretty late.

I will say it's amazing to me that we have so many students using Overleaf today at the undergraduate and sometimes even secondary level. I don't think I'd done anything other than write things out by longhand for most of my university career until that internship.

PN: You do have a PhD in mathematics, isn't it?

JLM: Yeah. I studied computer science as an undergraduate, and then I went on to do a PhD in engineering mathematics over at Bristol (UK).

PN: That's where your statistics part comes from. I read some of your papers and the ones on the game 2048 are very interesting. And some of the statistics there went beyond what I could understand easily.

To anybody listening to this interview, I do strongly recommend it; the game is addicting. It's called 2048. You can play it on the web, you can play it on an iPhone, or whatever.

But the papers by John are a little bit hard to get into. There is a higher level of math that's required to understand every step of it, but extremely interesting and also beautiful.

Let me ask you this: What moves you more, technology or coding?

JLM: Probably technology in general. I guess when I started out at Overleaf, I was pretty much doing the coding. But now Overleaf has grown to over 60 people, so my role has changed quite a lot. So I tend to work at a slightly higher level now. I occasionally get to touch the code, but mostly we have people who are professionals who do most of that now. So I try to operate a slightly higher level, looking at things like architecture and how the various bits fit together and long term vision for where we're going. I guess that's more on the technology side.

I feel like I should always say that my understanding of LaTeX is also very much a practitioner's

Paulo Ney de Souza

doi.org/10.47397/tb/43-2/tb134lees-miller-desouza

understanding. I did do Overleaf support, first line and second line and third line support for many years, and there I learned just how little LaTeX I really know. We later hired some very capable LaTeX experts like Lian Tze Lim and several more after her, who showed me that I know almost nothing, but I was still able to support some people who had basic LaTeX questions over the years.

PN: You were an experienced coder before you met TeX and LaTeX?

JLM: Yes.

PN: And that is what you used to start and build Overleaf?

JLM: Yes. I was already working as a software engineer, I guess going way back. I started it up pretty young. So when I was in middle school, after doing some Quick Basic, that was around the 2000 dot.com bubble, so anyone who could code anything could get a job. I actually got a part-time job even though I was like 13, writing [code]. I guess at that point it was Visual Basic rather than Quick Basic, but same idea.

So I started out there and then I did the computer science degree, in which mostly I studied the mathematical bits more than the practical ones, but I still worked quite a bit on the side. So I gained experience there. And I think depending on how far back in my history of blogs you go, there's a blog post from 2009 which I think is the earliest public record of Overleaf, or at least the idea behind Overleaf.

Back in the day there was this service called Etherpad, which is like a very early basic precursor to Google Docs. And so my collaborators and I, when I was a PhD student, were basically using that to write our papers. But then Google bought Etherpad and shut it all down. So before they'd done that, I'd actually written a bunch of scripts that sort of did crazy things like download the Etherpad and compile it, upload the PDF somewhere. So it kind of had all the components of Overleaf. But then as soon as Google came in and shut the whole thing down, I decided, well, it can't be too hard to start writing my own Etherpad thing. And so that's how Overleaf got started.

PN: You did meet John Hammersley before Overleaf, correct?

JLM: Right. So we were both working at a company called Advanced Transport Systems and we built the world's first computer-guided taxi system. It went on to be called the Heathrow Pod at Heathrow Airport, and was basically 20 computer-guided taxis that ran on their own roads. So this was very basic compared to the stuff that Google and Tesla, well, not Google any more, Waymo and Tesla are doing today, but we actually managed to get something into production in 2011, after that opened.

John and I met at the company. I think we were in the systems research division. It was a very small company, I think even smaller than Overleaf is now, but there was still a systems research division. We worked on simulation software mostly. One of the good things about the company was that it was very open. So we worked with a lot of people in academia and that was another reason that we needed a good collaboration tool, so that we could work on our papers together, with all the people at university and in the company. That's still something that people do on Overleaf today.

PN: What do you consider more important in terms of skills, to have competing, complementary or similar skills as your partner, in developing an idea like Overleaf?

JLM: You definitely benefit from having complementary skills. Pretty rare to find someone who can found a business as a solo founder, but they certainly do exist. There are just a lot of different aspects of a business that you need to cover. So the split between John H and I was always that John looked at more of the business side of things, more of the commercial side, and I looked after the technology side and that worked pretty well.

John is actually pretty technical. He doesn't always let on, but he was also fairly technical. But he could do, much more effectively than I, many of the commercial parts of the business, like talking to our early customers in the publishing industry, raising investment; he definitely led all of that. That also freed up some of my time to focus on initially building the products and the prototypes and all of that, and then eventually going to manage the engineering team because there's a lot of hiring required to go from two people in 2012 to 60-odd today, ten years later.

PN: What would you name as your biggest challenge right now at Overleaf?

JLM: I think we have quite a few growing pains. So every time you double in size, something stops working, is my experience. Communication just gets harder and harder as you get larger. I'm always amazed that companies with thousands of employees can work at all. It's pretty challenging, even with 60 employees, to keep everyone on the same page, aligned. Something that we are still working on, I would say. I think all companies still have to work at that, no matter how big they are.

Probably these days it's almost a cliche, but it's about communication and trying to get all these people with different skills to work together, and work together efficiently. We certainly benefit from having a very passionate group of people. Something I feel extremely lucky and proud of is that everyone at Overleaf is very passionate about either the technology or the impact that we have on making science communication a little bit faster and easier. And there's lots of people that just really like LaTeX and TeX. I feel it's a very special company to have so many people [who] are really passionate about the mission that we have.

PN: Is TeX Live a big stone on your path? TeX Live changes?

JLM: Well, we're extremely grateful to TeX Live. Without TeX Live we would really struggle to maintain any kind of compatibility between Overleaf and the offline world, and that's something that we definitely try to do. Ideally, if it works offline, it should work on Overleaf, and vice versa. We have very few things that are Overleaf-specific.

That said, yes, TeX Live is a challenging target. The fact that it's always a moving target is probably the biggest challenge we have because we have to cut a release and ship that to millions of people.

Right now, TeX Live 2022 is in the testing phase [at Overleaf]. So basically what we do is we take a snapshot of TeX Live sometime pretty soon after the official release and then we run it against all of our gallery projects. Overleaf has a template gallery where there are around 10,000 projects that people have submitted that are licensed appropriately for us to be able to use them for testing. We basically just run all of those and see what breaks. And if something very serious breaks, then we have to figure out how to patch our image to try to fix that without breaking too many other things. So, yeah, we definitely do find the TeX Live release process a bit of a struggle and we are in contact with the TeX Live team.

Usually by the time we find one of the problems with our gallery, though, we discovered it's already been reported by someone in the open source community. So it's pretty rare that we actually find a new bug, but we certainly hit most of the bugs that everybody else hits. So hopefully, if all goes well, I think it's in the final round of testing and the key guy, Eric, is on holiday for a week, but when he's back, I think we're going to hit the publish button and that will become available.

PN: Would you allow me to ask a question that I asked JH last time around?

JLM: Sure.

PN: Will we be able to use Overleaf on our iPhones and Android as a native app?

JLM: I can neither confirm nor deny that there are plans to do that, but a native app is definitely on our radar.

One of the things we're working on at the moment is that we've moved the entire editor, the actual sort of thing that you type into, from a piece of software called Ace to a newer piece of software called Code Mirror Six, which is a big project. So currently I think 20% of users can see the new editor because we roll out most things incrementally now, because if we roll out to everyone at once, we very quickly get overwhelmed if there's anything wrong with it. So that should have much better support on mobile. It uses a very different approach to actually making the editor work, which plays a lot better with mobile browsers.

So at least for this year, and probably next year, our focus is just on getting it working better on web browsers, on mobile devices. But I am sure that the day will come when we do make the jump into native apps. I just don't know when.

PN: Thank you very, very much. This was a wonderful conversation and hope to have you here back again sometime soon.

JLM: Great, thanks for having me.

PN: Thank you.

JLM: If anybody has any questions, I'm happy to stick around.

PN: Thank you. If you want to join the floor, just raise your hand and we still have time. So if you have any questions for John, raise your hand and we will bring you in.

But first, John, I don't know if you can talk about all of this, but how much of the company is remote and how much is it physically local?

JLM: Yes, I can definitely talk about that. It's in our job ads. We are now basically all remote.

We have staff in, actually I've lost count of the number of countries now, but most of our staff are in the UK, the US, some in Canada; we have a growing number in Germany, and places like France and Portugal. So we are all remote.

We're not a fully distributed team though, so we do try to constrain our hiring to a range of time zones. That means that people have overlap. So basically we have what are called core hours, which are 2:00 p.m. to 5:00 p.m., UK time. So if you're in the UK, you tend to have a slightly quieter morning and if you're in the US, you have a slightly quieter afternoon to get on with things, and then all the meetings in the core hours. That's how we manage that.

Frank Mittelbach (FMi): Can you hear me, John?

Your last statement about your gallery testing and everything made me wonder if we should talk with you about the possibility to align that with the LaTeX releases. Right now you probably have seen our sins in various respects, because every half year we have the LaTeX releases quite heavily sort of improving stuff. But also in corner cases breaking stuff, which is a natural thing if you have millions of users out there.

You may or may not know that we run a development version of LaTeX which is available to everybody. The intent is that developers and users can make use of that before we actually hit the street. I don't know exactly how you do your gallery testing, but if you have this as a sort of process, it might be quite helpful for everybody if that process could encompass running the development release at the late stage before we switch over. Then you would probably find bugs that otherwise will be found by you when you [take] the full release.

I think by the end of the day that would save you effort as well as saving everybody else pain. And if we have a sort of feedback loop this way, I think that could be very beneficial to everybody just to bring this up as a potential sort of alignment between our team and Overleaf there in the future.

JLM: Yes, we would be very happy to talk about that. As you say, we will hit the bugs anyway. I think we have talked about it at a previous TUG, and certainly something we're still interested in doing, setting up some kind of LaTeX CI, continuous integration, where we can just run it against our large sample of documents and report back on error rates, basically what we do internally now.

The other thing I'd say would be interesting is that we track performance across the TeX Live releases as well. One of the things that we see is that every year things tend to get slightly slower. So we would also be very interested in trying to set up some kind of benchmark test set because we're not sure that our gallery test is particularly representative as a benchmarking tool. But probably we could set up some sort of thing we would be more comfortable with as a benchmarking tool that could also alert on things like performance regressions, which at the moment I think there's not a lot of visibility into.

PN: You have a question from Jérémy on the chat.

JJ: Can you tell us about the computing power behind Overleaf? How many computing nodes, and what kind of nodes?

JLM: I can say a few things, and it ties into the slight performance decreases every year as more and more servers are required. Overleaf is hosted on the Google Cloud platform now. When we did the integration with ShareLaTeX in 2017, we were hosted on essentially every cloud service, which meant that if any cloud service was down, Overleaf had some kind of problem. So one of the things that I've been working on for the last five years is consolidating all of our hosting on Google Cloud.

The number of nodes, I don't think we give an exact number, but I can say it is many, many hundreds of cores, continuously compiling people's LaTeX; and there's also some fraction of that for running the service. But our LaTeX compilers are by far the most compute-hungry thing that we do.

JJ: Thank you.

PN: Thank you very much, John, for joining us, taking time out of your Sunday for this.

Welcome to TUG, and please come back and join us at other years.

JLM: Great. Thanks again for having me. Thanks, everyone. Have a good rest of the day.