

# *DEEP DIVE*

---

## *KIDS AND LINUX*





# The Kids Take Over

As with Linux, these kids are all about making things—and then making them better. They're also up against incumbent top-down systems they will reform or defeat. Those are the only choices.

*By Doc Searls*

It starts here, in the heart of Long Island, a couple dozen exits east of Queens. I saw it with my own eyes in [Mineola's Public Schools](#), where kids, led by a nonprofit called

**kidOYO** (“kid-oh-yo”), are learning to program in different languages on different devices and operating systems, creating and re-creating software and hardware, with fun and at speed. Their esteem in themselves and in the eyes of their peers derives from their actual work and their helpfulness to others. They are also moving ahead through levels of productivity and confidence that are sure to create real-world results and strip the gears of any system meant to contain them. Mineola’s schools are not one of those systems.

OYO means Own Your Own, and that’s what these kids are learning to do. In geekier terms, they are rooting their own lives online. They’re doing it by learning to program in languages that start with **Scratch** and progress through Python, Java, C# and beyond. They’re doing it on every hardware and software platform they can, while staying anchored to Linux, because Linux is where the roots of personal freedom and agency go deepest. And they’re doing it all in the spirit of **Linus’ book title**: *just for fun*.

With kidOYO, the heuristics go both ways: kidOYO teaches the kids, and the kids teach kidOYO. Iteration is constant. What works gets improved, and what doesn’t gets tossed. The measures of success are how enthused the kids stay, how much they give and get energy from each other, and how much they learn and teach. Nowhere are they sorted into bell curves or given caste-producing labels, such as “gifted” or “challenged”. Nor are they captive to the old report-card system. When they do take standardized tests, for example the college AP (advanced placement) ones for computer science, they **tend to kick ass**.

kidOYO is the creation of **the Loffreto family**: Devon and Melora, and their son Zhen, who is now 13. What started as a way to teach computing to Zhen turned into ways to teach computer science to every kid, everywhere. kidOYO’s methods resemble how the Linux kernel constantly improves, with code contributors and maintainers stamping out bugs and iterating toward ever-expanding completeness, guided by an equal mix of purpose and fun.

Before we met, I had assumed, from Devon’s writing style and deep knowledge



**Figure 1. Melora, Zhen and Devon Loffreto**

of stuff, that he was a gentleman perhaps of my own age, or even older. So I was surprised to find that he was not only a youngish guy, but also a New York state high school champion baseball and basketball player who went to college on a sports scholarship—also that he looked a stunt double for George Clooney.

I've also known for a long time that what kidOYO does is important. But my mind wasn't blown by it until I obeyed Devon's invitation to see their approach at work. That happened on Groundhog Day in February of this year. (An album of pictures I took on that visit is available on the [Linux Journal Flickr site here](#). Many of the links in this article go to captioned photos in that album.)

Mineola is about as prototypical as a middle-class New York suburban town can get. It's a two-square mile [village](#) of about 20,000 in the heart of Nassau County, located between Long Island's north and south shore and home to about 1.5 million people. The [Mineola Free Union School District](#), however, is anything but typical. I've never seen a public—or any—school system with its feet equally planted in the digital and the physical worlds, or as determined to help kids master both. For example, all three schools I visited had created social and hacker spaces, called Coding Centers, within their libraries. The books and the stacks still mattered, but so did the ability of kids to research, learn and teach together using computing and related gear, such as 3D printers and programmable robots.

Standing in the Coding Center at the Mineola Middle School, surrounded by kids doing amazing stuff on their Chromebooks, [Dr. Michael Nagler \(@naglersnotions\)](#), superintendent for the district, gave me the backstory on how kidOYO got involved:

Three years ago, my wife signed our son up for a coding class these guys were putting on. So I drive my son out there, and I'm watching what they're doing, and I'm impressed. I ask Dev, "Why aren't you in schools?" He says, "The schools won't talk to us." So I say, "Well, you're talking to one now." We worked to help adapt their platform for schools, starting with ours. And I mean *all* of ours. We jumped in the deep end, starting with the little kids and pushing it up through high school. And now we're on this three-year journey, so far, during which everything changes. Constantly. The little ones get the skills, and they roll up. Now I have to adjust my next level, and do it waaay faster than I have to with any other curriculum. Right now, for example, for the AP Computer Principles course in high school, they're doing the learning path for Hatch 1 and Hatch 2.

Later, when I asked Melora in an email what Hatch was, she replied, “Hatch is an app within OYOclass that uses the Scratch programming language. Here are two projects made in Hatch: [one by 10-year-old kid](#)[OYO Student ‘Lucy’](#) and [one by me](#).”

Dr. Nagler continued:

Meanwhile, my sixth graders are already finished with it. So by the time these sixth and seventh graders get to ninth grade, my expectation is that every student in the district is taking AP Computer Principles. That’s going to replace our Exploring Computer Science class. And then we build in connections. So we’re doing Arduinos here in the Middle School’s sixth grade, and simultaneously in ninth grade in the high school. Then, as the younger kids move forward, we’ll change the ninth grade setup.

Since [Maker Faire New York](#) is a great place for kids from everywhere to show off their maker chops (and where I first met the whole Loffreto family), I asked Dr. Nagler if they had plans for that. He responded, “We merge CS and computational thinking with making. We have a whole design and creative thinking framework tied to our mascot, the mustang. We make ways for the kids to conceptualize, design, iterate, prototype, test, refine, go, back, and build things.”

I asked, “How do you deal with the variety of kids who are already on this path, plus other kids who want to come in and need to catch up, and eventually have everybody in the school doing AP-level work on computers?”

He replied:

A couple ways. First, it’s not an elective. Here in Mineola, every kid has to do it. They also have to do it in their subject classes. So we tie a coding project to a curriculum project. Every grade has to do three a year. We also teach it both independently the OYO way, and in the existing the formal way, cycling kids through CS classes, for example here in this room. I think we’re unique in that we don’t want it to be a formal class. I want CS to be ingrained in everything we do.



**Figure 2. Dr. Nagler and His Brain**

I asked, “How do you see this scaling and spreading?” And Dr. Nagler said:

We constantly refine what we do so we can share it in ways that can be adopted by other districts. I’m a big open-source guy. Sharing is key. So, for example, I’m taking the kidOYO platform and building an open computer science curriculum in social space. The beauty of their platform is that it lets me build OER—Open Educational Resources—using their concept of learning paths, which we also work on together. Dev also built me a community that I can share with

an organization I belong to called the [League of Innovative Schools](#), which is a national organization. We can crowd-source content there. I built a sample curriculum unit I can push outside New York to other states. By crowdsourcing we already have a ton of content on there.

(Later, Melora clarified what’s happening here: “Dr. Nagler is building a repository of open curriculum of all subjects currently taught in school. The CS curriculum comes from kidOYO.”)

At this point, [Devon joined the conversation](#). “Tell Doc about [MC2](#).”

“Right. It stands for Mineola Creative Content, and it’s a video production studio. We do fun learning videos, which are a basis for the learning pathway here.”

The opening text on the [MC2 site](#) explains, “This community showcases open educational content and other materials from the Mineola School District....Our school district is dedicated to the #GoOpen movement, which supports sharing educational resources.”

“It’s all about #OER—Open Educational Resources—and open source”, Dr. Nagler explained. “We use the videos here in the district, and we throw them out to the world where everybody can use them.”

[Look up “Dr. Nagler” on YouTube](#), and you’ll find lots of them. He’s the star, as both a mentor and an animated character. There’s even one video where he talks with his own disembodied brain, which speaks through his signature goatee. He explained further:

An important context is that there is no central repository of educational materials in this country, because they’re all locked up by proprietary publishers. What we’re doing here is a way to get around that. And I have a lot of flexibility. I can market MC2 as a school district entity, and not worry about all the copyright restrictions. It’s all made to share.

I asked him, “What happens after these kids graduate?”

They’re going to change the world. That’s clear. We’re also all dealing with astronomical change in the technical environment along the way. Constantly. This makes everything very hard to predict. Look at my 2019 high school graduates. They started Kindergarten in 2006. Even from just 2006 to 2009, the technology advances were astronomical. And then look what happened in the next ten years. Huge. So if I start planning now for where Kindergarten kids will come out at the end of the next 12 years, I’m already lost. But if I trust the process we have in place already, I’ll be fine. We’re driving it, and the kids are driving it too. It’s a constant cycle.

I replied, “We also live in a world where giant companies are working to contain those kids’ agency inside corporate silos. Some of those silos also spy on everyone constantly. How do you deal with that?”

The common denominator is CS, and the flexibility within it. There’s freedom in that. I’m not going to force you to master, say, just one language. I’m going to get you on a platform where you can play with any and all of them, learn quickly and well, and apply whatever language you like toward building something. And because we’re merging the making and the coding, your next question will be, “What will this code *do*?” The answer is, computational thinking will always push you toward solving problems. If you look at the big picture, content already is readily available to every kid. And content has always been our specialty, as a school. But with CS, the kids learn to master that content in many ways. That’s key. Kids need to know and feel they’re on top of things, that they Own their Own. You can’t lock up that kind of confidence and competence.

I asked, “What about curricular necessities? The mandates that come down from the federal and state level?”

Dr. Nagler replied, “We’re still a public school, and we do have formalities. For example, here in New York every kid has to pass the [state Regents Exam](#). We teach

to that, but we also make sure there's no way a kid graduates without exposure to computer science."

My next question to him was "And you trust that's going to equip them, once they're out?"

It's more than that. Working with kidOYO, we've developed something that not only should be replicated everywhere, but needs to be. Here's the important thing: there aren't enough people who know computer science who can also teach it. So when you figure out a way to virtually do it, to scale the knowledge outward for everybody, it's a big deal. The investment I make here probably cost me one teacher's salary. But it scales to the whole district. In fact it's the only way to scale up computer science through schools, because the current credentialing system is too slow, too top-down, and formal training is too far behind the curve. The kids and their mentors are moving too fast for that.

Watching the kids, and listening to this, made me wish I could show it all to [John Taylor Gatto](#), possibly the most highly regarded (and often awarded) teacher in the history of New York. Gatto famously quit his job after 25 years in protest against what he listed as called the seven lessons he was actually paid to teach:

1. Confusion
2. Class position
3. Indifference
4. Emotional dependency
5. Intellectual dependency
6. Provisional self esteem
7. That you can't hide



**Figure 3. Jordan Chaver and Connor Scott, Co-hacking in the Coding Center at Mineola Middle School**

What I saw in both kidOYO's and Mineola's approaches were well crafted ways to fight all of that. Their systems are rigged so every kid progresses and every kid succeeds.

John Taylor Gatto died last October, but I hope his ghost was listening a few minutes earlier when Melora explained to me:

We have no lowest common denominator, because everyone succeeds. There are 12-year olds in this program that a 7th-grade teacher wouldn't look twice at in an ordinary classroom, but in fact are outstanding programmers. And choice is key. When Dr. Nagler brought this program to his schools, it wasn't just for a select few kids. He wanted it to be open to everybody. And everybody has the ability to choose anything they want. It's a totally different ecosystem from what you'll find anywhere else. And he's gracious enough to reach out to other school systems to

help them break down their own classroom walls. One of the things he preaches is that you have to believe. That's a requirement of being on the cutting edge. The failing forward principle works for everybody too. It's a model that works.

The spirit of helpfulness and failing forward also fosters kids' confidence that they can weigh in with solutions of all kinds. To show me how that works, Devon took me over to a table where Jordan Chaver and Connor Scott, a sixth-grader and seventh-grader, were working together on something. Devon said:

These two guys are your app builders. They came with us out to Stony Brook University for some of our software program there. Jordan pitched them on building an app on iOS, which he already knew how to do. But there was not a single mentor in the room who knew what Jordan was trying to do, because in university CS, they don't want to work in a closed environment. So we transitioned the challenge over to the web, because what we really needed was a web-based app with database functionality. So that's what these kids are building here. And there isn't just one app. There's a set of them. There's one they call Social Emotional. There's another called Class Dash.

Then Devon asked the boys to demo Class Dash. Connor pulled up a Chromebook, angled it toward me and said, "Let's say you have a research paper. One that's big and complicated. And you press Submit. Behind this you have something kind of like Dropbox, where you can share documents."

Devon explained, "They're sharing all their class assignments in a firewalled white-spaced environment where they don't have access to their emails. So this is a simple way of sharing inside that environment."

Connor continued:

You also have this five-character ID code. Jordan can type in the code, and he gets the same exact document. So can anyone else with the code. The idea is to share something with the class in a way that avoids complications. We're also in a class play,

*Once Upon a Mattress*, which is based on the *Princess and the Pea*. I'm the Prince and Jordan is the Wizard. So Jordan made this schedule for all the performances, where you can buy tickets, and so on.

On his Chromebook, Jordan showed me his page with the schedule next to a graphic of the play's title. He then gave Connor the five-digit code for access to the schedule, which then came up on Connor's Chromebook. (A picture of that is [here](#).)

Connor again: "Right now, I'm adding a way to lock a document. Let's say that Jordan is the teacher and he finds a spelling error in my document. I'll add a button you can click on to see if anybody has updated the document."

Jordan said:

Let me tell you more about Class Dash, which I did for Stony Brook. It's a student-teacher companion app. It has multiple uses, but the one that's currently available is called Schedule. It covers notes, teacher, room, and supplies. I play drums, so drumsticks are an example of supplies. I also have Instant Messaging Teacher. The idea is, if you have a homework question, instead of emailing the teacher and getting a response the morning after, the teacher gets a push notification on their phone.

Class Dash will first hit the market in April as an iOS app, because that's Jordan's plan. Other versions will come after that.

[Joseph Malone, also 12, is at the same table, hacking AI algorithms](#). Devon said, "Joseph here is spinning up his own virtual machine and generating algorithms to train his AI to run his scripts. He's going into [OpenAI](#), playing with AI algorithms, modifying them, and putting them to use. It's neat stuff, and it's also huge."

Melora told me Joseph is also helping out by volunteering a stream of challenges, solutions and badges for kidOYO courseware. "He does all the work himself, and makes it open and available to everybody."



**Figure 4. Joseph Malone**

“We’re fully networked here,” Devon added. “No need for back-end support.” Meaning no external corporate dependencies. kidOYO and its participants—learners (they aren’t called students), mentors (they aren’t called teachers), parents, schools—all work together and for each other, as a “community of communities”.

They’re also not moving at the speed of anybody’s clock or anybody’s class. Although they’re sure to change the world, that’s not the goal. In fact, there is no long-term goal. The journey is truly the reward, and the journey is called the *learning path*. That’s what matters. It’s not seen, or built, as a way to plow through the status quo, even though that’s one of the things it does. Neither Mineola nor kidOYO want to

burden kids with anything other than the need to master their digital worlds and to advance their mastery constantly.

The Middle School was the second one we visited in Mineola. The first was Hampton Street School, which is Pre-K to 2nd grade. There we saw clusters of five- and six-year-old **girls** and **boys** in the library's **Coding Center**, hacking away on school-issued tablets using **Scratch**, which is free (as in both liberty and cost), open source and runs on anything. They were doing this both **by themselves** and **collaboratively**.

With kidOYO, all the kids know they are working to expand both their own skills and those of other kids. There also are rewards along the way, such as on-screen fireworks and badges. After a bit of working on their own, the kids' work is **shown on a screen for review by each other and Melora, their mentor**. (The learner/mentor relationship is central to the kidOYO system and practiced in the Mineola school system as well.) Devon later explained what was going on: "Melora was reviewing the process of getting challenge submission feedback from mentors, as well as introducing them to a new app called Sprite Editor that we recently released for kids to create art they may want add to their Scratch, Python or Web-based projects. Often it's their own video game character art."

When **one boy failed** a particular challenge, he embraced it, knowing that FAIL means "First Attempt In Learning". **Three girls** came over to help the boy out. It was interesting to watch how they knew their job wasn't to jump in with the right answer, but to help the boy learn what he didn't know yet, so he would have the satisfaction of succeeding for himself. This was far more sophisticated and mature than I normally would expect of 2nd-grade kids. Instead, I would have expected kids that age to show off what they knew or one-up each other. But that's not how the kidOYO approach works.

Have you ever played the **red/black game**? It tends to be taught in self-improvement retreats and workshops to show there's more to be gained from cooperation than from competition. My point in bringing it up is that it's damned hard to teach adults how to deal with each other in ways that are as empathetic,



**Figure 5. Kids Helping Each Other “Fail Forward”**

helpful and vanity-free as what I saw as normal behavior among these little kids.

At Hampton Street, Devon spent most of his time working with a 2nd-grader

named William Ponce, who clearly was grooving on what he was doing. Later, Devon wrote to explain what was going on:

[Here is William Ponce's portfolio](#). Every kid has one. You can see badges he has earned. If you click on one of his “Mastery Badges”, you will see the “Learning Pathway” that he navigated in earning it, displayed as evidence in the badge. Clicking on the micro badges will also show you the badges earned on his way to the mastery badge.

[In this photo](#), you see William earning his first Mastery Badge. Since we left that class, you can see he has earned two more already!

Our third stop was Mineola High School, which has a [Fab Lab](#) and manufacturing facility. “We actually source product from them”, Devon told us on the way over. “For [our store](#). Coding is the underlying infrastructure, but it’s applied everywhere.”

The Fab Lab is beyond impressive. It’s as big as a lumber yard and has lots of machinery, materials and students making stuff. Ken Coy, one of the five teachers who collaborate to run the lab, explained:

We do it all. Welding, electronics, coding, Arduino, hand tools, computer tools. We bring it all together here. We have all the old traditional tools that were around in wood shop days—drill press, band saw, lathe, tools for sanding—plus all the new stuff that’s both manual and computer controlled. Large format printers, laser cutters...

When I asked him about Linux, he brought me over to the shop’s [Linux CNC \(Computer Numerical Control\) computer, running Ubuntu](#) and attached to a [Probotix](#) controller and a router (not a network router, but a [powered woodworking tool](#) that cuts with bits or blades). In the design class space, Andrew Woolsey ([@WoolseyDesigns](#)) showed me a CNC controlled laser cutter where the students were tracing, carving and printing out parts for art projects, signs and much more (which occupied students working on adjacent tables).



**Figure 6. Devon Loffreto Mentors William Ponce**

He also showed me a printer as wide as a piano churning out student portraits and posters of amazing quality, including ones for the Mineola Robotics Team (@mineolarobotics), which is always competitive (or so it appeared, given the awards and posters hanging on the shop wall).

I don't often see stuff that makes me wish I was 14 again, but Mineola High School did the job. Walking around the Fab Lab, the library and the halls, I didn't see a kid who wasn't upbeat and engaged, or a teacher who wasn't the same.

To me, however, this isn't just about education. Or learning. It's about a sea change in the world, caused by digital technology in general and Linux in particular. And it's not a small one. As sea changes go, this one is on the scale of



**Figure 7. Linux in the Fab Lab**

**Snowball Earth** or maybe larger.

Not long ago, I was talking with **Joi Ito**, who runs the **MIT Media Lab**, about historic precedents for what we might call our species' digital transition: the one by which we become digital as well as physical animals. Was it as big as the Industrial Revolution? Movable type? Writing? Speech? Walking on two feet? Joi said, "I think it's the biggest thing since **oxygenation**."

Oxygenation caused life as we've known it since then. What is the digital transformation causing now?

Marshall McLuhan taught that our tools are extensions of our selves, and that they shape us after we shape them. He also said every useful new technology “works us over completely”. That’s what’s happening in our new digital age, and it’s still just beginning.

At this early stage, it’s easy to take a dystopian view of what becoming digital does to kids. It is also easy to take a utopian one. Both are extreme outcomes that surely won’t happen. But what will?

Aristotle said there were four causes: material (what something is made of), efficient (what makes it happen), final (the purpose) and formal (the *form* or design of the result).

These kids’ learning paths are full of material, efficient and final causes. To them, those are computer programs (material), programming (efficient) and rewards at every step (final). But the formal cause I saw behind them, the *design* of OYO itself, is a great leap forward and outward in the useful work of individuals and the societies they make.

There will be downsides. One of the ways new technologies work us over, McLuhan said, is with bad outcomes. We already can see some, such as the social isolation that comes from staring at glowing rectangles all the time. Every parent I know laments the degrees to which their children are lost in the phones and tablets they carry everywhere, and how they can so easily hurt each other through unkind things said at safe distances in the physical world and zero distance in the networked one.

But the OYO approach maximizes positive social interaction by making it constructive for everybody. OYO doesn’t work unless people are good to each other and good to themselves—and by making stuff constantly and being creative.



**Figure 8. Our Future's in Good Hands**

If this approach spreads, and I expect it will (mostly because the old industrial education system is better off adopting than competing with it), the hands in which we are leaving the world will be good ones. ■



**Doc Searls** is editor-in-chief of *Linux Journal*, where he has been on the masthead since 1996. He is also co-author of *The Cluetrain Manifesto* (Basic Books, 2000, 2010), author of *The Intention Economy: When Customers Take Charge* (Harvard Business Review Press, 2012), a fellow of the *Center for Information Technology & Society (CITS)* at the University of California, Santa Barbara, and an alumnus fellow of the *Berkman Klien Center for Internet & Society* at Harvard University. He continues to run *ProjectVRM*, which he launched at the BKC in 2006, and is a co-founder and board member of its nonprofit spinoff, *Customer Commons*. Contact Doc through [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).

Send comments or feedback  
via <https://www.linuxjournal.com/contact>  
or email [ljeditor@linuxjournal.com](mailto:ljeditor@linuxjournal.com).