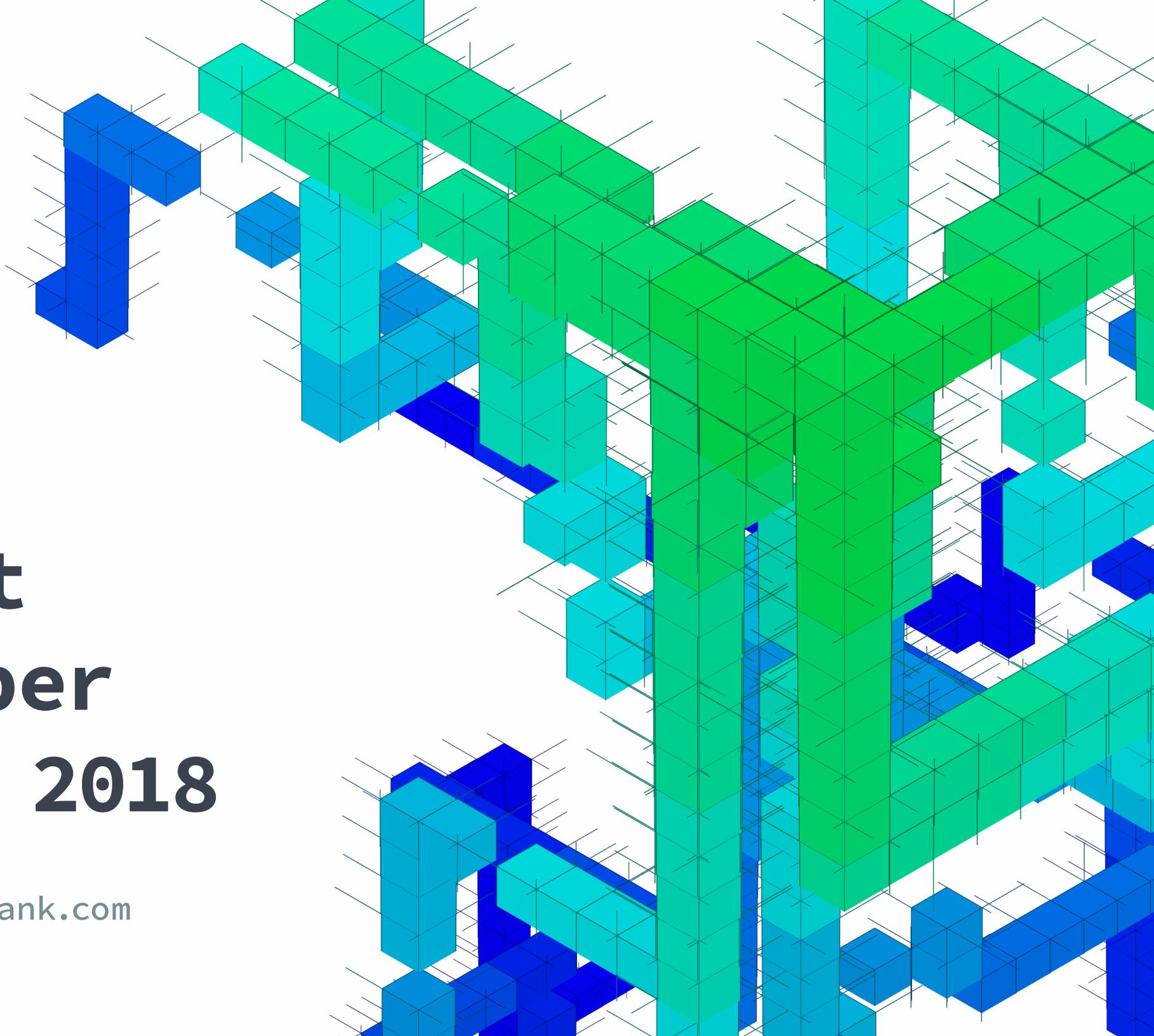




Student Developer Report 2018

research.hackerrank.com



Skills First

How do you identify young talent that will have the greatest impact?

For most companies, the standard practice is to target select, prestigious schools, invest in a prime spot during seasonal career fairs, and collect as many resumes as possible to build an entry-level talent pipeline. It's tempting to look at blunt proxies—like GPAs and internships at brand name companies—to create a candidate shortlist.

As someone who has been on both sides of the university recruiting table, I can tell you that this method is not sustainable. First off, in the age of digital transformation, the demand for developers won't be able to keep up with such a narrow strategy. If everyone's fighting for talent in the same pools of talent, hiring will be exponentially more difficult. Secondly, if your goal is to make strong hires, grades and alma mater aren't reflective of true ability.

Back in college, my co-founder Hari Karunanidhi and I used to spend hours competing in coding contests. We had solid fundamentals and were really strong developers, but didn't necessarily have the pedigree that recruiters were looking for.

I was fortunate that recruiters from Amazon happened to come to my campus for a coding competition that I was participating in—I was able to demonstrate my skills. Many other developers may not be as lucky.

It's time to focus the conversation on skill. To dive deeper on understanding the state of student developer skills, we surveyed over 10,000 developers across universities worldwide on how they're learning, what they're learning, and what they look for in a job. For instance, we learned that even though 76% of tech students are pursuing a degree in computer science, 65% of all student developers say they are at least partially self-taught. And nearly one third of all developers said they're completely self-taught.

Self-directed learning is the norm among developers; so when companies focus on hiring based on proven skill instead of prestigious degree or GPA, a massive pool of overlooked talent opens up.

We hope you find this useful as you prepare your next university recruiting season.

Feel free to [contact us](#) if you have any questions.



Vivek Ravisankar

Co-founder & CEO

HackerRank



research@hackerrank.com
www.hackerrank.com

USA:
+1-415-900-4023

India:
+91-888-081-1222

UK:
+44-208-004-0258

A degree is not enough to learn how to code

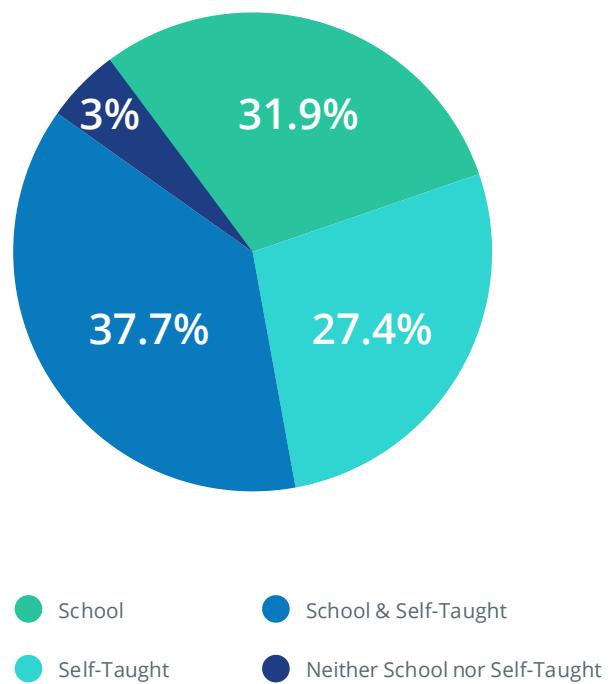
Although most computer science students are learning software development in universities, over half of all student developers say they're partially self-taught. And nearly one third of all student developers say they're completely self-taught.

This prevalence of self-taught knowledge means two things: First, computer science programs lag behind the pace at which technology evolves. For skills that are growing in the industry today—like [the latest frameworks](#)—students have to rely on self-teaching to learn.

Second, self-teaching is ingrained in the developer DNA. Developers have an insatiable thirst for learning. On average, students are learning 6 languages, while more senior developers are learning 4. Developers in the Asian Pacific region are the most ambitious, with the average student planning to learn 7 programming languages!

Their thirst for learning makes it clear that Computer Science (CS) degrees shouldn't be the primary measure of student developer skills. Instead, teams need to look beyond school performance—to personal projects, portfolio, skills assessments, and the like—to accurately evaluate their skills.

How did you learn to code?



research@hackerrank.com
www.hackerrank.com

USA:
+1-415-900-4023

India:
+91-888-081-1222

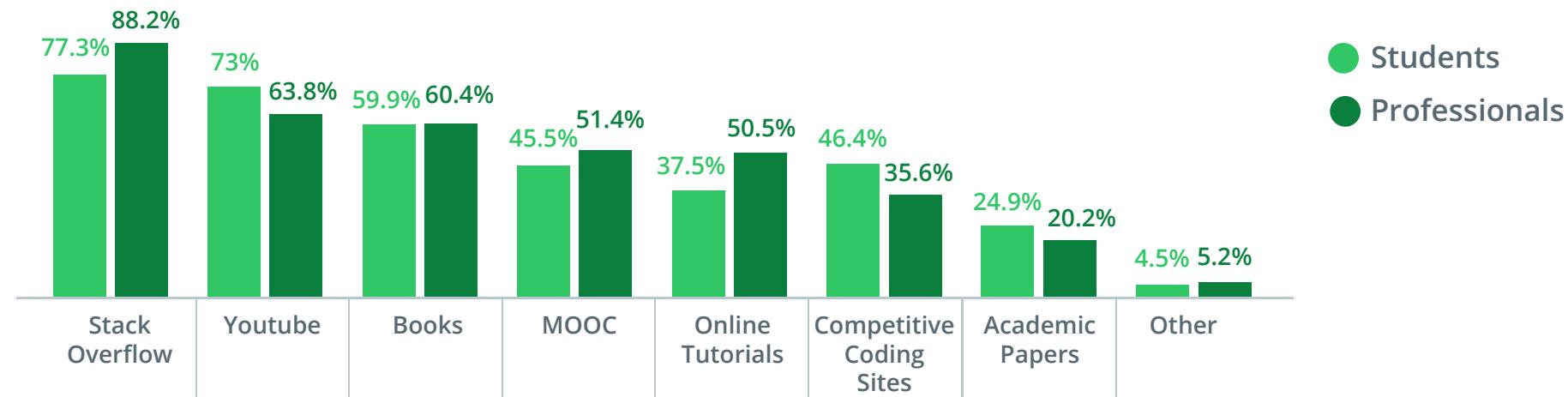
UK:
+44-208-004-0258

Students rely more on YouTube than professionals

Developers have traditionally flocked to Stack Overflow as a means of learning new skills or working through problems. But that could be changing with new generations.

University students today are showing less interest in Stack Overflow compared to professional developers. Instead, YouTube is starting to become more favorable as a learning tool for developers with the next generation of developers. We found that 73% of students use YouTube, compared to only 64% of professional developers.

Besides HackerRank, which of these platforms do you use to learn how to code?



Outside of the classroom, the face of education is shapeshifting quickly. The popularity of these more customizable, self-directed mediums is the beginning of a larger generational shift led by Generation Z (those born in the mid 90's – early 00's), who make up the majority of today's college students.

This finding supports recent research by [Pearson & Harris Poll](#), which found that Generation Z (ages 14-23) preferred YouTube and video to learn versus other applications of learning.



research@hackerrank.com
www.hackerrank.com

USA:
+1-415-900-4023

India:
+91-888-081-1222

UK:
+44-208-004-0258

Globally, demand for JavaScript is outpacing student expertise

While 48% of employers say they need JavaScript skills, only 42% of student developers worldwide say they know the language. When we dug deeper, we found there is a regional difference. The gap is particularly wide in India and Canada. Meanwhile, the US and UK have the greatest population of JavaScript developers.

And that skills gap could be traced back to CS curriculums. By and large, JavaScript isn't taught in standard computer science curriculums. In fact, of US News and World Report's top 10 CS programs, zero explicitly include JavaScript in their core curriculum; skilled students must rely on internships, personal projects, or mentorship to learn the language. It's why students that are more likely to teach themselves to code (including American and British students) have a better grasp on JavaScript.

But CS programs are unlikely to incorporate JavaScript any time soon. Their focus on coding theory (versus application) leaves little room for practical, industry-driven languages like JavaScript. And for the fraction of programs that do want to incorporate JavaScript, feasibility presents a challenge. [Almost half of JavaScript developers](#) feel the JavaScript ecosystem is changing too fast—so even those hungry to include it may struggle to keep their curriculums up to speed.

Which languages do employers need versus student developers know?

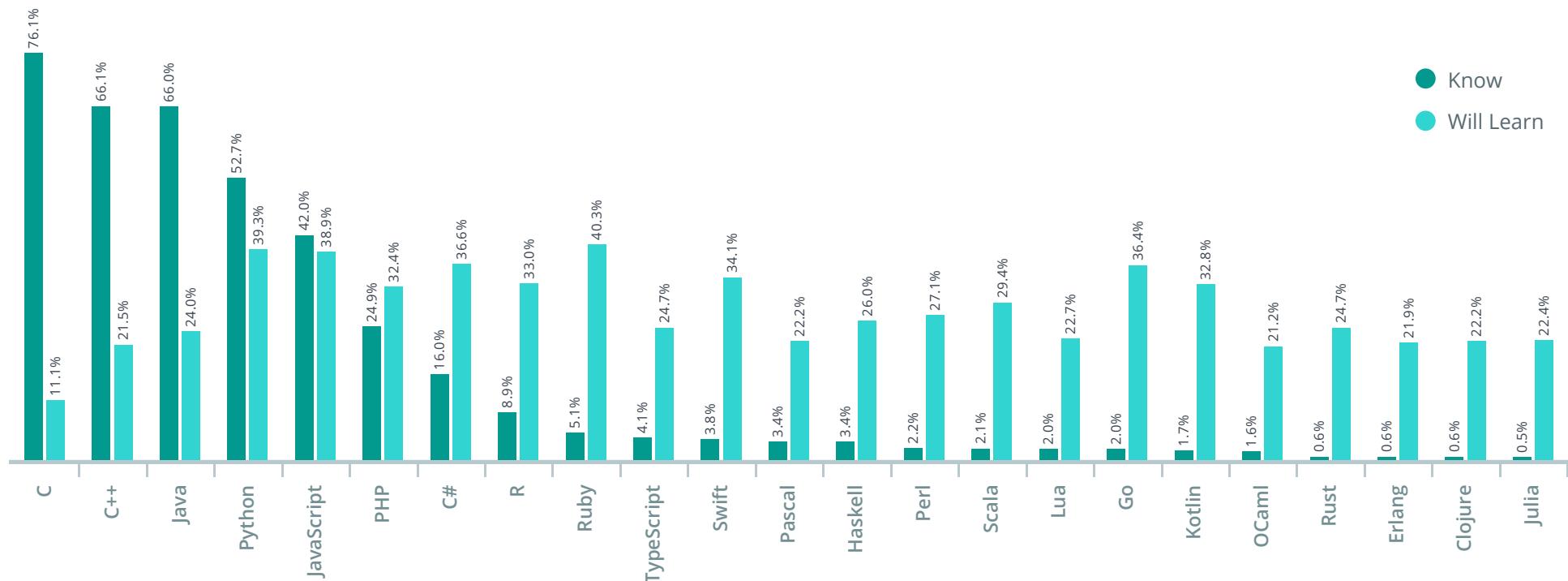


But Ruby, Python, & JavaScript top students' to-do lists

Despite the JavaScript gap, there's good news for employers: students are keen on learning it. In fact, JavaScript is one of the top 3 skills they want to learn next, coming in 3rd after Ruby (1st priority) and Python (2nd priority).

Having a strong foundation of classic lower-level languages, like C, helps developers pick up new languages more easily. And as we saw earlier, students are highly motivated to learn independently, since the majority are at least partially self-taught. Awareness of the massive opportunities found in JavaScript and JavaScript-frameworks can help bridge the gaps.

Which languages do students know versus plan to learn next?



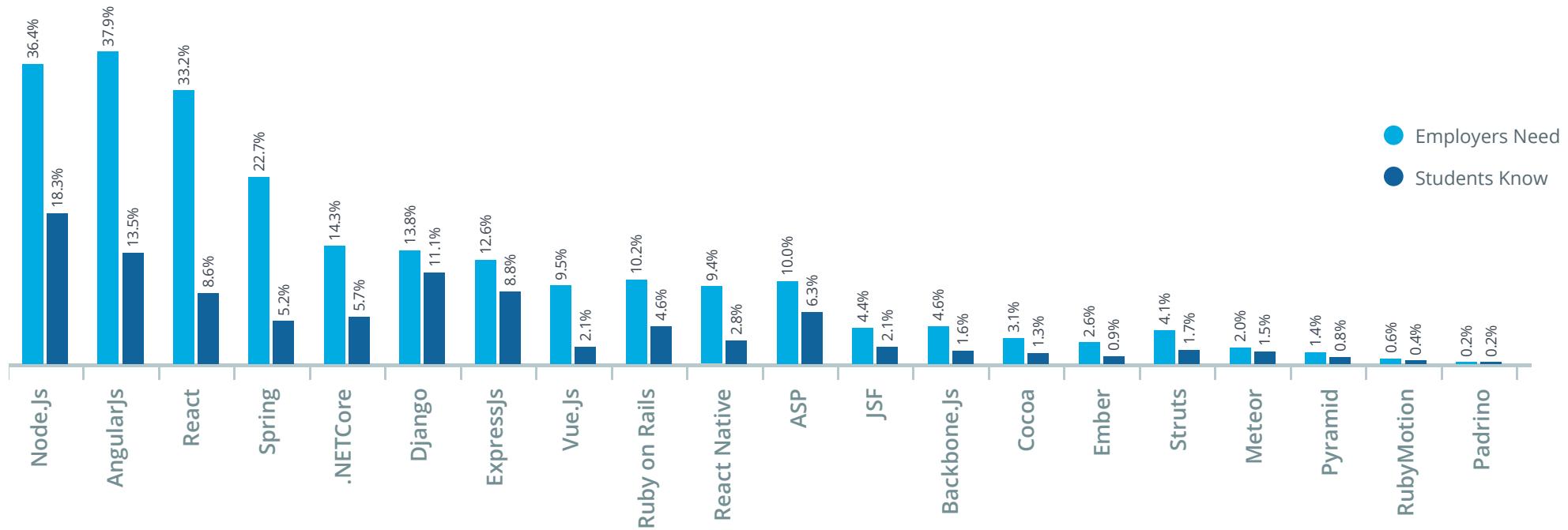
Frameworks are generally learned on the job

Students are underprepared to leverage the frameworks employers need, with the biggest knowledge gaps residing in AngularJS, React, Node.js, and Spring. No individual country's students could meet even half of employers' framework demands.

And again, this discrepancy traces back to fundamentals-focused CS curriculums; in most CS programs, students learn to build knowledge of theories and fundamentals, not job skills. The reason is multi-faceted: university curriculums have neither the industry feedback, nor the agility, nor the resources to stay in sync with industry trends.

Frameworks are best learned in execution, so students should work on projects that allow them to learn building on frameworks. And hiring teams should adjust their expectations, given the tremendous skill gap, and invest in training new grads and early talent.

Which frameworks do employers need versus student developers know?



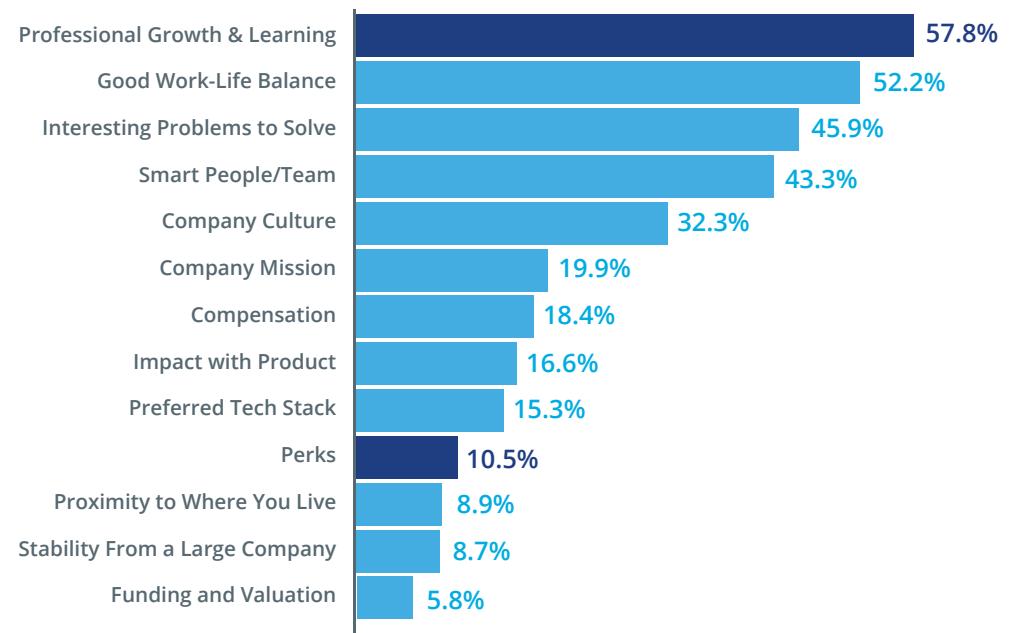
Growth opportunities appeal 5x more than perks

To woo students globally, employers should focus first and foremost on students' #1 job priority: professional growth and learning.

Good work-life balance and interesting work are also important to students, landing at priority numbers 2 and 3, respectively. Highlighting the expertise of the existing team, mentorship opportunities, and the impact of the team's work can help employers stand out here. Emphasizing opportunities for work-life balance—like remote working options and flexible schedules—can also help.

Ultimately, students are hungry to learn, and to grow their careers. Capitalize on their enthusiasm by offering opportunities to own interesting, challenging projects; they'll appreciate the opportunity to grow their skills, and seasoned teammates will get a chance to flex their mentorship capabilities.

Globally, what do student developers want most in a job?



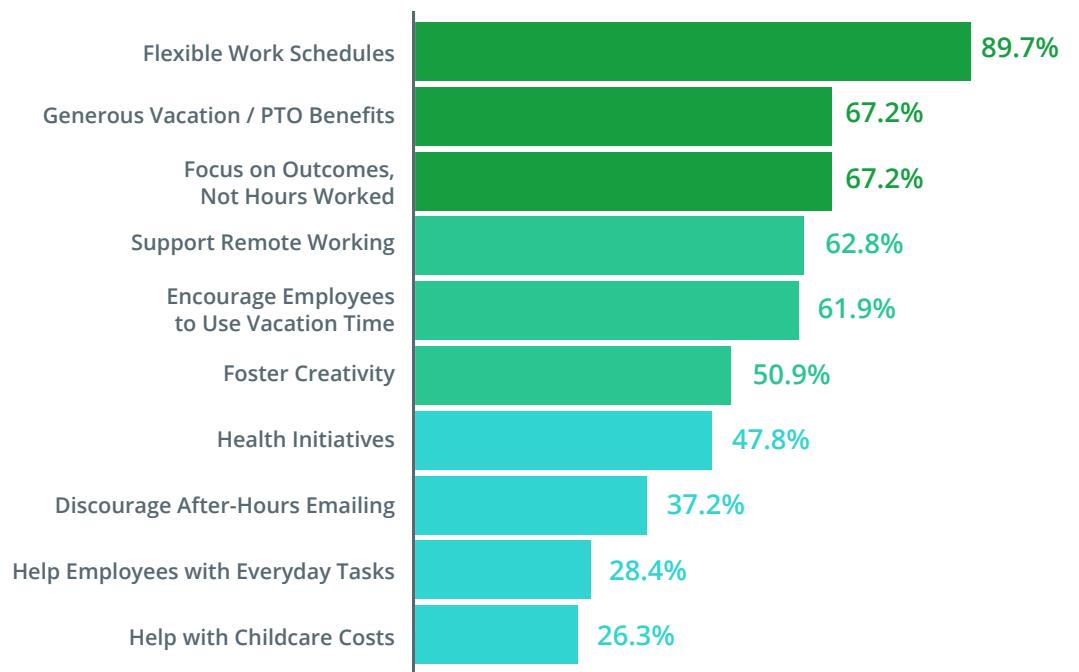
And in the U.S., work-life balance tops growth opportunities

Globally, students are aligned on job priorities, but with one exception: the U.S.

U.S. students are significantly more interested in company culture than students from any other country, and are also the only student group that identify “work-life balance” as their #1 priority. More specifically, U.S. students are drawn to flexible work hours, a focus on outcomes (not hours), and generous PTO benefits.

To speak to these students, employers shouldn't just talk about work-life balance: they should actively encourage it. Even when younger generations are offered balance opportunities, **they fear what coworkers will think** if they take advantage of them. Openly encouraging a culture of balance will help them feel at ease.

What does work-life balance mean to US students?



More Research



Tech Recruiting
Report
May 22, 2018



Women in
Tech Report
March 1, 2018



Developer
Skills Report
January 23, 2018



The United
States
of Developers
July 26, 2017

Methodology

We conducted a study of developers to identify trends in developer education, skills and hiring practices. A total of 10,351 student developers completed the 10-minute online survey from October 16 to November 1, 2017.

The survey was hosted by SurveyMonkey. We recruited respondents via email from our community of developers, and through social media sites.

Tests of significant differences were conducted at the .01 level (99% probability that the difference is real, not by chance). Percentages may not always add to 100% due to rounding.



research@hackerrank.com
www.hackerrank.com

USA:
+1-415-900-4023

India:
+91-888-081-1222

UK:
+44-208-004-0258



research.hackerrank.com

Match Every Developer to the Right Job

USA:

+1-415-900-4023

India:

+91-888-081-1222

UK:

+44-208-004-0258

research@hackerrank.com

www.hackerrank.com