**Title**

JavaScript is for Girls

**Permalink**

https://escholarship.org/uc/item/5sd1x9kf

**Journal**

LOGIC, 1(1)

**Author**

Posner, Miriam

**Publication Date**

2017-03-15

*Unedited version of an article published in the spring 2017 edition of LOGIC Magazine (https://logicmag.io/)*

# JavaScript is for Girls

by Miriam Posner

Technology has a gender problem, as everyone knows. The underrepresentation of women in technical fields has spawned legions of TED talks, SxSW panels, and women-friendly coding bootcamps. I've participated in some of these get-women-to-code workshops myself, and I sometimes encourage my students to get involved. Recently, though, I've noticed something strange: the women who are so assiduously learning to code seem to be devaluing certain tech roles simply by *occupying* them.

It's not always obvious to outsiders, but the term "technology sector" is a catch-all for a large array of distinct jobs. Of course there are PR, HR, and management roles. But even if we confine ourselves to web development, technical people often distinguish among "front-end," "back-end," and "full-stack" development. The partition between the two "ends" is the web itself. There are people who design and implement what you see in your web browser, there are people who do the programming that works behind the scenes, and there are people who do it all.

In practice, the distinction is murky: some developers refer to *everything* user-facing as the front end, including databases and applications, and some developers use front-end to mean only what the user sees. But while the line shifts depending on who you're talking to, most developers acknowledge its existence.

I spoke to a number of developers who confirmed something I'd sensed: for some time, the technology industry has enforced a distinct hierarchy between front-end and back-end development. Front-end dev work isn't real engineering, the story goes. Real programmers work on the back end, with serious programming languages. Women are often typecast as front-end developers, specializing in the somehow more feminine work of design, user experience, and front-end coding.

Are women really more likely to be front-end developers? Numbers are hard to pin down. Most studies consider the tech sector as a single entity, with software engineers lumped together with HR professionals. A 2016 StackOverflow user survey showed that front-end jobs —"Designer," "Quality Assurance," and "Front-End Web Developer" — were indeed the top three titles held by women in the tech industry, although that survey itself has some problems. We need better numbers, as feminist developers have been saying for years, but it also doesn't seem like a huge stretch to take developers at their word when they say that front-end development is understood to occupy the girlier end of the tech spectrum. Front-end developers, importantly, make about $30,000 less than people in back-end jobs like devops engineers, according to the salary aggregation site Glassdoor.

The distinction between back and front wasn't always so rigid. "In the earliest days, maybe for the first ten years of the web, every developer had to be full-stack," says Caroline Ada Ehmke, a Chicago-based developer who has worked on various parts of the technology stack since 1993. "There wasn't specialization." Over time, web work professionalized, and by the late 2000s, Ehmke says, the industry began to stratify, with developers who had CS degrees (usually men) occupying the back-end roles and self-taught coders and designers slotting into the front.

For many people who are teaching themselves to code, front-end work is the lowest-hanging fruit. You can "view source" on almost any webpage to see how it's made, and any number of novices have taught themselves web-styling basics by customizing WordPress themes. If you're curious, motivated, and have access to a computer, you can, eventually, get the hang of building and styling a webpage.

Which is not to say it's easy, particularly at the professional level. A front-end developer has to hold thousands of page elements in her mind at once. Styles overwrite each other constantly, and what works on one page may be disastrous on another page connected to the same stylesheet. Front-end development is taxing, complex work, and increasingly it involves full-fledged scripting languages like JavaScript and PHP.

"Serious" developers often avoid acknowledging this by attributing front-end expertise not to mastery but to "alchemy," "wizardry," or "magic." Its adepts don't succeed through technical skill so much as a kind of web-whispering: feeling, rather than thinking, their way through a tangle of competing styles. "There's this perception of it being sort of a messy problem that you have to wrangle with systems and processes rather than using your math-y logic," says Emily Nakashima, a full-stack developer based in San Francisco. That's not true, of course; nothing on a computer is any more or less logical than anything else. But perhaps it's easier to cast women in a front-end role if you imbue it with some of the same qualities you impute to women.

The gendered attributes switch as you travel to the back of the stack. At the far end, developers (more often "engineers," actually) are imagined to be relentlessly logical, asocial sci-fi enthusiasts; bearded geniuses in the Woz tradition. Occupations like devops and network administration are "tied to this old-school idea of your crusty neckbeard dude, sitting in his basement, who hasn't showered in a week," says Jillian Foley, a former full-stack developer who's now earning her doctorate in history. "Which is totally unfair! But that's where my brain goes."

The brilliant but unkempt genius is a familiar figure in the history of computing — familiar, but not immutable. Computing was originally the province of women (a fact innumerable articles and books have pointed out but which still seems to surprise everyone every time it's "revealed"). The bearded savant of CS lore was the result of the field's professionalization and increasing prestige, according to the computing historian Nathan Ensmenger. "If you're worried about your professional status, one way

to police gender boundaries is through educational credentials," says Ensmenger. "The other way, though, is genius. And that's something I think nerd culture does really well. It's a way of defining your value and uniqueness in a field in which the relationship between credentials and ability is kind of fuzzy." And "genius," of course, is a strongly male-gendered attribute; just look at teaching evaluations.
When programming professionalized, women got pushed out. Marie Hicks, a computing historian who's looked closely at this phenomenon, explains that as programming came to be viewed as more important to national and corporate welfare, hiring managers began associating it with a specific set of skills. In the British case, Hicks's specialty, a good programmer was an ultimate systems-thinker, able to see and synthesize the big picture. In the US, as Ensmenger and others have documented, the best programmers were purportedly introverted chess nerds, obsessed with details, logic, and order. (There's very little evidence that these characteristics actually make a good programmer.)

The traits of a "good programmer" differed by country, but they were universally male-gendered, enforced by hiring managers and other programmers who sought to replicate their own characteristics — not consciously, for the most part, but simply because the jobs were important. Conscientious hiring managers wanted to bet on qualities everyone agreed were indicators of success. "The people with more prestige in a culture are favored for all sorts of things, including jobs," says Hicks. "If you have a job that you want to fill, you want to get the best worker for it. So in more prestigious fields, employers are looking for those employees that they think are the best bet. This tends to attract men who are white or upper-class into these more desirable jobs."

People often think that as a profession matures, it gets more complex, says Hicks, and thus edges women out because it demands higher-level skills. But "historically, there's very little to bear that out," says Hicks, who has uncovered multiple incidents of women programmers training, and then being replaced by, their male counterparts.

The case of the female front-end developer is flipped in the other direction — it's a feminizing subfield, rather than a masculinizing one. But it's governed by many of the same market forces that edged women out of programming in the first place: prestige accrues to labor scarcity, and masculinity accrues to prestige. Front-end jobs are easier for women to obtain, and feminized jobs are less prestigious. In turn, the labor market generates its own circular logic: women are front-end developers because they're well-disposed to this kind of labor, and we know this because women are front-end developers.

No one says this any of this explicitly, of course, which is why the problem of women in technology is thornier than shoehorning women onto all-male panels. The developers I spoke to told me about much more subtle, very likely unconscious incidents of being steered toward one specialization or another. Two different women told me about accomplished female acquaintances being encouraged to take quality assurance jobs, currently one of the least prestigious tech gigs. Ehmke told me about a friend who applied for a back-end developer position. Over the course of the interview, the job somehow morphed into a full-stack job — for which Ehmke's friend was ultimately rejected, because she didn't have the requisite front-end skills.

And everyone can rattle off a list of traits that supposedly makes women better front-end coders: they're better at working with people, they're more aesthetically inclined, they care about looks, they're good at multitasking. None of these attributes, of course, biologically inheres to women, but it's hard to dispute this logic when it's reinforced throughout the workplace.

Once you're cast as a front-end developer, it can be challenging to move to different parts of the stack, thus limiting the languages and development practices you're exposed to. "Particularly in Silicon Valley, there's a culture of saying developers should always be learning new things," says Nakashima, the San Francisco-based full-stack developer. Front-end specialization "can be a place that people go to and don't come back from. They're working on these creative projects that are in some ways very interesting, but don't allow them to move to an area of the stack that's becoming more popular."

Viewed from one angle, the rise of get-girls-to-code initiatives is progressive-minded and feminist. Many people involved in the movement are certainly progressive feminists themselves, and many women have benefited from these initiatives. But there are other ways to look at it, too. Women are generally cheaper, to other workers' dismay. "Introducing women into a discipline can be seen as empowerment for women," says Ensmenger. "But it is often seen by men as a reduction of their status. Because, historically speaking, the more women in a profession, the lower-paid it is."

An influx (modest though it is) of women into the computing profession might be helping to push developers to make distinctions where they didn't exist before. "As professions are under threat, stratification is very often the result," says Ensmenger. "So you take those elements that are most ambiguous and you push those, in a sense, down and out. And down and out means they become more accessible to other groups, like women." But these roles are also markedly distinct from the main work of software engineering —which is safely insulated from the devaluing effect of feminization, at least for the time being.

Hicks, the computing historian, can't stand it when people tout coding camps as a solution to technology's gender problem. "I think these initiatives are well-meaning, but they totally misunderstand the problem. The pipeline is not the problem; the meritocracy is the problem. The idea that we'll just stuff people into the pipeline assumes a meritocracy that does not exist." Ironically, says Hicks, these coding initiatives are, consciously or not, betting on their graduates' failure. If bootcamp graduates succeed, they'll flood the market, devaluing the entire profession. "If you can be the exception who becomes successful, then you can take advantage of all the gatekeeping mechanisms," says Hicks. "But if you aren't the exception, and the gatekeeping starts to fall away, then the profession becomes less prestigious."

My students are always so excited that they're "learning to code" when I teach them HTML and CSS, the basic building blocks of webpages. And I'm happy for them; it's exhilarating to see, for the first time, how the web is built. Increasingly, though, I feel the need to warn them: the technology sector, like any other labor market, is a ruthless stratifier. And learning to code, no matter how good they get at it, won't gain them entrance to a club run by people who don't look like them.