

# Falsehoods Programmers Believe About Names

[Readers have translated this essay. You're welcome to translate it into any language; I'd appreciate you sending me an email so I can link to it.

- プログラマの抱いている名前についての誤謬
- 戳这里看中文翻译

]

John Graham-Cumming wrote an [article](#) today complaining about how a computer system he was working with described his last name as having invalid characters. It of course does not, because anything someone tells you is their name is — by definition — an appropriate identifier for them. John was understandably vexed about this situation, and he has every right to be, because **names are central to our identities, virtually by definition**.

I have lived in Japan for several years, programming in a professional capacity, and I have broken many systems by the simple expedient of being introduced into them. (Most people call me Patrick McKenzie, but I'll acknowledge as correct any of six different “full” names, any many systems I deal with will accept precisely none of them.) Similarly, I've worked with Big Freaking Enterprises which, by dint of doing business globally, have theoretically designed their systems to allow all names to work in them. **I have never seen a computer system which handles names properly and doubt one exists, anywhere.**

So, as a public service, I'm going to list assumptions your systems probably make about names. **All of these assumptions are wrong.** Try to make less of them next time you write a system which touches names.

1. People have exactly one canonical full name.
2. People have exactly one full name which they go by.
3. People have, at this point in time, exactly one canonical full name.
4. People have, at this point in time, one full name which they go by.
5. People have exactly N names, for any value of N.
6. People's names fit within a certain defined amount of space.
7. People's names do not change.
8. People's names change, but only at a certain enumerated set of events.
9. People's names are written in ASCII.
10. People's names are written in any single character set.
11. People's names are all mapped in Unicode code points.
12. People's names are case sensitive.
13. People's names are case insensitive.
14. People's names sometimes have prefixes or suffixes, but you can safely ignore those.
15. People's names do not contain numbers.
16. People's names are not written in ALL CAPS.
17. People's names are not written in all lower case letters.
18. People's names have an order to them. Picking any ordering scheme will automatically result in consistent ordering among all systems, as long as both use the same ordering scheme for the same name.
19. People's first names and last names are, by necessity, different.
20. People have last names, family names, or anything else which is shared by folks recognized as their relatives.
21. People's names are globally unique.
22. People's names are *almost* globally unique.
23. Alright alright but surely people's names are diverse enough such that no million people share the same name.
24. My system will never have to deal with names from China.
25. Or Japan.
26. Or Korea.
27. Or Ireland, the United Kingdom, the United States, Spain, Mexico, Brazil, Peru, Russia, Sweden, Botswana, South Africa, Trinidad, Haiti, France, or the Klingon Empire, all of which have “weird” naming schemes in common use.
28. That Klingon Empire thing was a joke, right?
29. Confound your cultural relativism! People *in my society*, at least, agree on one commonly accepted standard for names.
30. There exists an algorithm which transforms names and can be reversed losslessly. (Yes, yes, you can do it if your algorithm returns the input. You get a gold star.)
31. I can safely assume that this dictionary of bad words contains no people's names in it.
32. People's names are assigned at birth.
33. OK, maybe not at birth, but at least pretty close to birth.
34. Alright, alright, within a year or so of birth.
35. Five years?
36. You're kidding me, right?
37. Two different systems containing data about the same person will use the same name for that person.
38. Two different data entry operators, given a person's name, will by necessity enter bitwise equivalent strings on any single system, if the system is well-designed.
39. People whose names break my system are weird outliers. They should have had solid, acceptable names, like 田中太郎.
40. People have names.

This list is by no means exhaustive. If you need examples of real names which disprove any of the above commonly held misconceptions, I will happily introduce you to several. Feel free to add other misconceptions in the comments, and refer people to this post the next time they suggest a genius idea like a database table with a first\_name and last\_name column.

Originally written: June 16, 2010

## ABOUT THE AUTHOR



Patrick McKenzie (patio11) has run several software companies, led national shadow digital infrastructure, worked in the tech industry, and writes about financial infrastructure.

## WHO AM I?

My name is Patrick McKenzie (better known as patio11 on the Internets.)

Twitter: [@patio11](#) HN: [patio11](#)

## BITS ABOUT MONEY

I write [Bits about Money](#), a monthly-ish newsletter on the intersection of tech and finance.

## COMPLEX SYSTEMS

I host the [Complex Systems](#) podcast, a weekly conversation about the technical and human factors underlying infrastructure.

**OLDER · [VIEW ARCHIVE \(575\)](#)**

## Detecting Bots with Javascript for Better A/B Test Results

I am a big believer in not spending time creating features until you know customers actually need them. This goes the same for OSS projects: there is no point in overly complicating things until “customers” tell you they need to be a little more complicated.

(Helpfully, here some customers are actually capable of helping themselves... well, OK, it is *theoretically* possible at any rate.)

**NEWER**

## **Running Apache On A Memory-Constrained VPS**

Yesterday about a hundred thousand people visited this blog due to my post on [names](#), and the server it was on died several fiery deaths. This has been a persistent issue for me in dealing with Apache (the site dies nearly every time I get Reddited — with only about 10,000 visitors each time, which *shouldn't be a big number on the Internet*), but no amount of enabling WordPress cache plugins, tweaking my Apache settings, upgrading the VPS' RAM, or Googling lead me to a solution.

© Patrick McKenzie (Kalzumeus Software, LLC) 2006 - 2025.

Shoutout to Elle Kasai for the [Shiori Theme](#).