

Forum:[Hack Forums](#)

Thread: [So I've got this mental bias against frameworks](#)

littleblueguy24 - June 11, 2022, 6:42 am

Can someone more knowledgeable than I explain the line between "you need to learn how do this yourself." And "there's a framework/library for that."

rihosedemon - June 11, 2022, 8:35 am

(Jun 11, 2022 06:42 AM)littleblueguy24 Wrote: [\[CITE\]](#) Can someone more knowledgeable than I explain the line between "you need to learn how do this yourself." And "there's a framework/library for that."

I am total newbie but as far as i know you need to understand the code behind the scenes of framework before jump to it so when a bug occurs to find easily the solution.

Disease - June 11, 2022, 10:17 am

(Jun 11, 2022 06:42 AM)littleblueguy24 Wrote: [\[CITE\]](#) Can someone more knowledgeable than I explain the line between "you need to learn how do this yourself." And "there's a framework/library for that."

The two- using frameworks or libraries and learning how to do things yourself- are not mutually exclusive. You can do both, I assure you of that. And you'll find that professional (doing this as a career) developers do both. Why? Because there's no reason to reinvent the wheel every time.

That said, good frameworks encapsulate the problem. That's what they're supposed to do. Now this has a lot of positives. Well-written frameworks or libraries put forth best practices, utilize appropriate design patterns, and in general just do a better job of handling the problem than you would on your own. The downside is this is you don't need to learn how to tackle that problem at a granular level.

That downside is what some developers (and they tend to be vocal about this) get hung up on.

But at the end of the day who cares? If you're a new developer you need to be focused on learning the language and simply how to be a programmer (which has nothing to do with the language). Once you're established, and have some experience under your belt, it's all about productivity and efficiency.

Lindros - June 11, 2022, 6:57 pm

(Jun 11, 2022 06:42 AM)littleblueguy24 Wrote: [\[CITE\]](#) Can someone more knowledgeable than I explain the line between "you need to learn how do this yourself." And "there's a framework/library for that."

Why would you reinvent the wheel when there is a likely a community driven, existing wheel that is better than anything you could do yourself?

Why would you waste your own time to make, what presumably would be, a fuckin' worse product?

Are you better than a massive community of developers collaborating together?

Null - June 11, 2022, 7:51 pm

I think there's no general rule when it comes to learning stuff.

You're basically comparing a bottom-up approach to a top-down approach, some people are comfortable starting from a higher level and they tear down abstraction layers as they go.

Others may feel comfortable building up from the first principles.

It's also very subject to the topic at hand as some topics are easier tackled in certain ways.

(Jun 11, 2022 06:57 PM)Lindros Wrote: [\[CITE\]](#) Why would you reinvent the wheel when there is a likely a community driven, existing wheel that is better than anything you could do yourself?

Why would you waste your own time to make, what presumably would be, a fuckin' worse product?

Are you better than a massive community of developers collaborating together?
I'm pretty sure some areas are best approached by reinventing the wheel.

Computer architecture, for example, is best approached by building a simple computer.

Practically speaking though, you don't start from scratch in the real world/industry unless you know what you're doing.

N3xtSolutions - June 12, 2022, 3:35 am

(Jun 11, 2022 06:57 PM)Lindros Wrote: [\[CITE\]](#) Why would you reinvent the wheel when there is a likely a community driven, existing wheel that is better than anything you could do yourself?

Why would you waste your own time to make, what presumably would be, a fuckin' worse product?

Are you better than a massive community of developers collaborating together?
Sometimes, yes. Even the mighty cpp ISO committee ended up releasing horrible(Performance wise) library functions such as accumulate and max.

But the whole point is that you want to know what the library/framework does under the hood so you don't run screaming to the senior dev about it when it doesn't work as it should.

I agree that modern frameworks/compilers are so much better than average joe with a keyboard.

Disease - June 12, 2022, 7:36 am

Something I didn't discuss in my last post- while most frameworks and libraries give you tools to tackle more complex problems, some exist to streamline common, everyday tasks. Those fitting the latter description may get in your way of learning the language and tackling common problems within that domain.

For example, in web development: jQuery. New developers coming up in web development (when jQuery was still stupidly popular) that picked up jQuery from the rip would often find themselves learning JS through the library. So they weren't learning the language, per se, but the library itself. These developers, unless they took it upon themselves to dig deeper, may have found themselves not knowing how to actually target an element in native JavaScript.

Conversely, picking up React right as you start JavaScript? That's not going to prevent you from learning the language. After all, React tackles a more complex and specific problem, and thus isn't encapsulating the language. If anything, React, like many frameworks across different languages, may push a new developer into learning complexities about the language sooner. That can be both a good and a bad thing.

sahara77 - July 16, 2022, 12:34 pm

Frameworks is needed...

You not have to reinvent the wheel !

SIM-574702 - July 16, 2022, 1:07 pm

You need to know basics about frameworks but once you need something that has a standard and lots of edge cases, it becomes really hard to implement yourself.

Just think about implementing CSV serializer and deserializer. Basic case for deserialization is `line.split(separator)` and it works in most cases. Problems start once the content of a line contains separator. What do you do once that happens? You surround that column with quotation marks in serializer. That also works fine until content contains quotation mark.

That's just an example of something you might think is easy to do on your own until you get into edge cases. If you want more examples, try to write or just think about how you'd write email validator or url validator. After you're done, check existing implementations to see how many cases you got wrong.

Dialog - July 17, 2022, 3:48 am

Whenever you're coding you will in practice always be relying on code written by others. The question is only to what extent. Even if you are building your own framework for executing business logic, I bet you will at the very least be invoking behaviour in your language of choice's standard library. There is really no point in trying to take one definite side in the question you're asking, because it all depends on context. The only important bit is that you will always need to understand the trade-offs, functional and non-functional, of whatever code you build on top on.

I see it as my job to write code in a way that is maintainable. I will pick whatever approach I expect to require least resources (compute and human) to run and maintain in the long run. That also means that even if I could write a very bomb web framework that does all the project needs, I will probably prefer picking an established one for that language (think: Ruby on Rails, Laravel, Django, et cetera). Even if that's only so that when new developers start working on the project, they don't need to figure out what the fuck I did three years ago that I probably forgot about by now, if I'm even working at the same place.

Sajuuk - July 18, 2022, 2:06 am

Seems everyone else has summed it up: don't reinvent the wheel.