

Help with programming

It's critically important to be able to choose the right tutorials on the arduous journey of learning programming. Not only are there tons of great and cool books in our field, but every day developers from all over the world publish dozens and hundreds of articles dealing with specific problems.

In such a chaos, it's very hard to find the most needed, fundamental knowledge, having it easy to understand and learn any other technology in a short period of time. In this article, we will look at the main sources of such fundamental knowledge: those references where you should start learning any technology. But be prepared: the suggested resources are much harder to apply than the conventional ones. But if you follow this advice, the speed of learning will increase many times over ;-)

Documentation

The first thing you should do is to google the official website of the technology and study carefully everything it says. The docs should always be your first point of contact with any library, programming language, framework or any other tool. Don't google "getting started with ruby on rails," don't search for "how todo list on react.js," and forget queries like "install elasticsearch." Remember, the technology is best known by its developers, and if they're good, they've already taken care of good documentation.

Github Issues, Github Pull requests, and other channels of communication with developers

Ok, so you've opened the official documentation, did everything like it, but the damn logstash still won't start. So you'll ask, "Well, is it time to google everything?"

No, it's not. Your next step is to look at the open and closed Issues on github, look at the pooling requesters there. If your problem was not caused by your inattention, either you already have a ticket for it in the project repository, or you should create one. At the same time you will get information on how to solve the problem from the developers of the problem themselves. Many major technologies (like Elasticsearch) even have separate forums and chats where you can ask your question. And you'll practice your English one more time!

[Help with programming homework](#)

[Jump to the source code!](#)

Okay. The project's documentation is terrible, there's been no activity on [GitHub](#) for a couple of years and/or the authors aren't responding to your questions. What do you do then? That's right: open the source code of that library (for programming languages it's a little more complicated, but still possible) and debug the bug you found yourself.