Theory: External resources

8 minutes 0 / 5 problems solved

You have learned how to write simple lines of code. But chances are that eventually, you'll run into a problem that every developer faces from time to time: what should I do if my code doesn't work at all, works too slowly, is badly written or I just don't know what to do next?

The first thing that comes to mind is Google. Don't be afraid to use it! Some people think that developers must know all about the tools they use, but in reality, no-one can know and remember everything. Even the most skilled developers often google things they don't know. Also, putting the problem into words usually helps you find a solution.

Now that you know googling is helpful, let us introduce you to some more specific resources that have saved many coders from nervous breakdowns and confusion. Note them down: you are sure to find them useful.

§1. Documentation

Let's face it: programming is complicated. Every programming language, framework, or technology has a lot of its own nuances. Moreover, technologies are always changing, with new versions replacing the old. So the first thing to do if you have chosen a specific language or technology is to look at the **documentation**. Technical documentation is a complete description of the technology which explains its functionality and how to use it. Programming language documentation usually also contains a description of syntax and language elements and even includes a simple tutorial.

No matter which language you are working with, documentation is easily available. Here is where you can find documentation for a few basic languages:

- Python 3: https://docs.python.org/3/
- Java: https://docs.oracle.com/en/java/javase/12/
- Kotlin: https://kotlinlang.org/docs/reference/

It is considered a good tone to write complete and readable documentation, so most of the programming tools' docs are pretty comprehensive, with each function or method described, syntax nuances noted and examples included. Always look at the docs of the tool you are using, especially when you are not familiar with it: it will save you a lot of time.

However, reading the documentation can turn out to be hard. That is because documentation, albeit a very useful resource which contains everything about the technology, is written in dry technical language and is loaded with information. It's like reading texts in a language you're trying to learn: at first, it's hard, but when you get used to it, it seems much easier. Moreover, finding the information you need in the documentation is a separate skill and takes practice and patience.

Don't be afraid if you are finding it hard to read the documentation – it's a challenge all of us face. But what if your question is so abstract or complicated that reading the docs don't help? Here comes another powerful resource!

§2. Stack Overflow

stackoverflow.com is the Bible of every developer, the largest question-and-answer site for programmers. Stack Overflow contains almost every question about programming ever asked. If you google some programming-related questions, it's very likely that the first link will lead to this site. In fact, some programming environments even have the option "Search Stack Overflow" that pops every time you get an error! Here's an example from *Google Colaboratory*, a powerful programming environment:

Current topic:

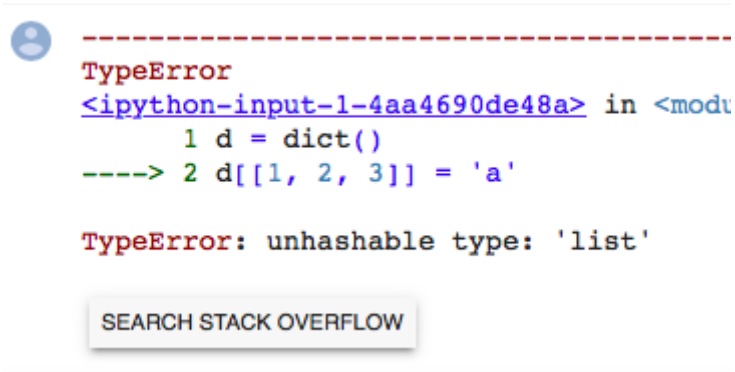
External resources

Topic is required for:

Introduction to Android

Basic project structure

Table of contents:



(of course, the site contains questions and answers about almost every programming language, not only Python)

So, Stack Overflow is the first resource you can turn to if your code doesn't work and reading the docs isn't helping, except Google and your own brilliant mind, of course. If you haven't found the answer to your problem, you can go ahead and ask a question, maybe you'll be the first person who ever asked that! Community and peer help are a very important part of learning, so don't be afraid to ask.

§3. Geeksforgeeks

Enjoyed the "algorithmic" section in our Knowledge map? Well, there is more computer science for you! And if you haven't read it, definitely put it on your list; meanwhile, here's a resource that would help you get a better grasp of computer science topics.

GeeksForGeeks.org is a portal devoted to computer science. It contains all the information about **algorithms and data structures**; it also touches on a lot of coding problems and questions that one can encounter at job interviews. Most information on this site is programming language-independent, related more to the math used in computer science than language-specific features. Solving problems and puzzles described on this site is a good way to get a better understanding of data structures and improve your coding and analytical skills. Also, if you've got a more theoretical question, there is a good chance that you will find an answer at GeeksForGeeks.

And again, if you don't understand something, don't be afraid to ask!

§4. Conclusion

As you see, you don't have to be alone with your code that won't work for whatever reason: there are many great sources out there that can help. If you wonder how to handle exceptions in Java, try StackOverflow, but "how do I reverse a linked list" is a good question for GeeksForGeeks. The last fundamental advice is this: use your brain and don't panic. Finding and correcting your errors might be frustrating, but once you get a hang of it, it becomes easier and starts seeming more like a routine than a catastrophe.

Remember: there is no such thing as "stupid question", as any question brings you a bit closer to your goal. Learn, improve and benefit from the existing resources: it's a smart move.

Report a typo

405 users liked this theory. **5** didn't like it. **What about you?**

Start practicing

Comments (7) Hints (0) Useful links (1) Show discussion