

# First Time Going Through Coding Interviews?

Here're some common questions and answers

## What's the interview process like at a tech company?

Good question. It's actually pretty different from most other companies.

Here's what it's like to interview for a programming job (</interview-process-at-tech-companies>).

## Do I need to know this "big O" stuff?

Big O notation (</big-o-notation-time-and-space-complexity>) is the language we use for talking about the efficiency of data structures and algorithms.

Will it come up in your interviews? Well, it depends. There are different types of interviews.

There's the classic algorithmic coding interview, sometimes called the "Google-style whiteboard interview." It's focused on data structures and algorithms (queues (</concept/queue>) and stacks (</concept/stack>), binary search (</concept/binary-search>), etc).

That's what our full course (</upgrade>) prepares you for. It's how the big players interview. Google (</google-interview-questions>), Facebook (</facebook-interview-questions>), Amazon (</amazon-interview-questions>), Microsoft (</microsoft-interview-questions>), Oracle (</oracle-interview-questions>), LinkedIn (</linkedin-interview-questions>), etc.

For startups and smaller shops, it's a mixed bag. Most will ask at least a few algorithmic questions. But they might also include some role-specific stuff, like Java questions ([/java-interview-questions](#)) or SQL questions ([/sql-interview-questions](#)) for a backend web engineer. They'll be especially interested in your ability to ship code without much direction. You might end up doing a code test or pair-programming exercise instead of a whiteboarding session.

To make sure you study for the right stuff, you should ask your recruiter what to expect. Send an email with a question like, "Is this interview going to cover data structures and algorithms? Or will it be more focused around coding in X language." They'll be happy to tell you.

If you've never learned about data structures and algorithms, or you're feeling a little rusty, check out our [Intuitive Guide to Data Structures and Algorithms](#) ([/data-structures-and-algorithms-guide](#)).

## **Which programming language should I use?**

Companies usually let you choose, in which case you should use your most comfortable language. If you know a bunch of languages, prefer one that lets you express more with fewer characters and fewer lines of code, like Python or Ruby. It keeps your whiteboard cleaner.

Try to stick with the same language for the whole interview, but sometimes you might want to switch languages for a question. E.g., processing a file line by line will be far easier in Python than in C++.

Sometimes, though, your interviewer will do this thing where they have a pet question that's, for example, C-specific. If you list C on your resume, they'll ask it.

So keep that in mind! If you're not confident with a language, make that clear on your resume. Put your less-strong languages under a header like 'Working Knowledge.'

## **What should I wear?**

A good rule of thumb is to dress a tiny step above what people normally wear to the office. For most west coast tech companies, the standard digs are just jeans and a t-shirt. Ask your recruiter what the office is like if you're worried about being too casual.

## **Should I send a thank-you note?**

Thank-you notes are nice, but they aren't really expected. Be casual if you send one. No need for a hand-calligraphed note on fancy stationery. Opt for a short email to your recruiter or the hiring manager. Thank them for helping you through the process, and ask them to relay your thanks to your interviewers.

---

Want more coding interview help?

Check out **[interviewcake.com](https://interviewcake.com)** for more advice, guides, and practice questions.