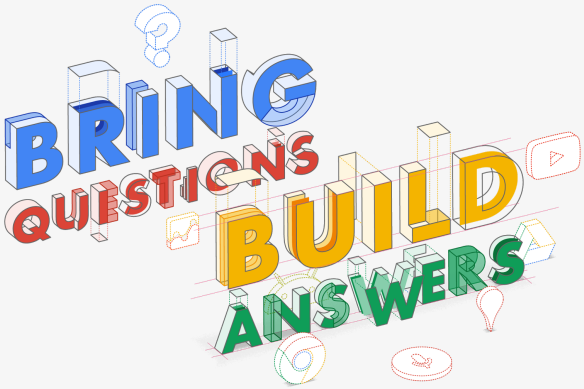

Google Interview Prep Guide

Developer Relations

What's Developer Relations (DevRel)? Developer Relations is the team behind all the rich, technical content on developers.google.com, as well as the folks you're likely to see speaking at Google I/O or the person you might be following on Google+.

Developer Advocates are focused on supporting the 3rd party developers who are building applications and businesses on Google's platforms. They are passionate evangelists for Google technologies as well as vocal champions for developer interests within Google. This is a position for engineers who love connecting with developers and speaking publicly about cutting-edge technologies at conference panels, user groups, blogs and with the press.

Developer Programs Engineers grow and support the developer community by teaching and supporting developers around the world. They write API docs, sample apps, client libraries, tutorials and blog posts. They also actively participate in developer forums and support queues to help developers with coding problems encountered when using our APIs and other developer products.

Why Google? Impact. Google is and always will be an engineering company. We hire people with a broad set of technical skills who are ready to tackle some of technology's greatest challenges and make an impact on millions, if not billions, of users. At Google, our team connects with developers and speaks publicly about cutting-edge technologies on conference panels, at user groups, on blogs and with the press. Chrome, Android, App Engine, HTML5 as well as our core Google Apps and Ads APIs are just some of the platforms you promote and support.

Interview Prep

Coding: You should know at least one programming language really well, preferably C++, Java, Python, Go, or C. Make sure to check out our Google code style guides.

Algorithms: Approach the problem with both bottom-up and top-down algorithms. You will be expected to know the complexity of an algorithm and how you can improve/change it. Algorithms that are used to solve Google problems include sorting (plus searching and binary search), divide-and-conquer, greediness, recursion dynamic programming/memoization, or algorithms linked to a specific data structure. We recommend discussing or outlining the algorithm you have in mind before writing code.

Data structures: You should study up on as many data structures as possible. Data structures most frequently used are arrays, linked lists, stacks, queues, hash-sets, hash-maps, hash-tables, dictionary, trees and binary trees, heaps and graphs. You should know the data structure inside out, and what algorithms tend to go along with each data structure.

Presentation: You may be required to give a 20-25 minute presentation about a technology of your choice. The presentation is a chance for us to understand your technical depth and knowledge of the subject while we assess your speaking skills. Ideally we'd like to see you present on something relevant to the challenges you would face at Google, treat it as if you were presenting at Google I/O. Please leave some time for Q&A and remember to make the presentation engaging and interactive!

Reflection: Reflect on your background and how it relates to the job description. Think through questions like, Why Google? Why are you qualified for this specific role? Why are you interested in it? Be sure to review the job postings minimum and preferred qualifications and role responsibilities, and be prepared to discuss how your background and skills align with those qualifications.

General Interview Tips

Explain: We want to understand how you think, so explain your thought process and decision making throughout the interview. Remember we're not only evaluating your technical ability, but also how you approach problems and try to solve them. Explicitly state and check assumptions with your interviewer to ensure they are reasonable.

Clarify: Many of the questions will be deliberately open-ended to provide insight into what categories and information you value within the technological puzzle. We're looking to see how you engage with the problem and your primary method for solving it. Be sure to talk through your thought process and feel free to ask specific questions if you need clarification.

Improve: Think about ways to improve the solution you present.It's worthwhile to think out loud about your initial thoughts to a question. In many cases, your first answer may need some refining and further explanation. If necessary, start with the brute force solution and improve on it — just let the interviewer know that's what you're doing and why.

Ask Questions: At the end of the interview, your interviewers will ask you if you have any questions about the company, work environment, their personal experiences, etc. This is your chance to learn more about the role, the projects and the type of work you'll be doing. Take advantage of the opportunity and be sure to bring questions.

Resources

Books
Cracking the Coding Interview
Gayle Laakmann McDowell
Programming Interviews Exposed: Secrets to Landing Your Next Job
John Mongan, Eric Giguere, Noah Suojanen, Noah Kindler
Programming Pearls
Jon Bentley
Introduction to Algorithms
Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein

DevRel Resources
What is Developer Relations?
Why does Dev Rel matter?
Core Competencies

About Google
Company - Google
The Google story
Life @ Google
Google Developers
Open Source Projects
Github: Google Style Guide

Interview Prep
How we hire
Interviewing @ Google
Candidate Coaching Session:Tech Interviewing
CodeJam: Practice & Learn
Technical Development Guide