Ultimate Guide to

Technical Interviews

CAREER KARMA.

Table of Contents

Introduction

No matter what process a company uses, this guide is here to help you succeed in your interview.

You have written a good resume and cover letter, applied for tons of jobs, and caught the attention of a good company. Now you've been invited in for an interview. What do you do? The interview is undoubtedly one of the most important parts of the hiring process: It allows a company to assess how you work in different environments, and helps them to gain a firmer understanding of your technical skills.

But the exact process a company may use to interview candidates will be different depending on their size. Smaller companies will have leaner hiring processes because they are still growing. Larger companies may have longer hiring processes because they have stronger internal policies around hiring new candidates. If you interview for a startup, you may only have to do one or two technical interviews, or maybe you will do a take-home task instead, or in addition to, a final technical interview. Larger companies, though, may start with a phone screen, then do a few on-site technical interviews.

No matter what process a company uses, this guide is here to help you succeed in your interview. In this guide, we'll explore what you can expect in technical interviews, and help you prepare for the specific types of questions you may encounter when you are in the interview room.

What is a Technical Interview?

Technical interviews usually come toward the end of the hiring process. After you have passed the resume and cover letter screen, and completed a phone screen, you may be invited in for one of these technical interviews. During the interview, you'll be asked a series of questions and presented with a few challenges that you are expected to solve. The interview is your opportunity to show a company that you actually have the skills you need to do the job, rather than having the ability to tell someone that you have those skills.

The purpose of the interview is not to trick you or give you incredibly complex problems to solve. Instead, the interview is designed to help you solve the types of problems you may encounter on the job. If you are applying for a developer operations (DevOps) job, you may be expected to solve a problem related to managing a cluster of servers at scale. This is likely a problem the company is facing, and they want someone who has the necessary skills to solve that type of problem.

There are a few components of a technical interview:

- Technical phone screen
- Take-home test
- On-site interview

But, as I mentioned earlier, some companies may have different processes. Facebook, for example, has an initial technical screen then jumps straight into an on-site interview which lasts for one day. Startups may skip the take-home test and only conduct one on-site interview if they are trying to hire a lot of people quickly.

Who You'll Be Talking With

Again, the details of who you talk with will vary depending on the company, but there are usually two possible scenarios:

- At a small company: members of the engineering team, or perhaps the CTO. All interviewers will be contributors to product development and may be your boss if you get hired.
- At a large company: recruiters, HR representatives, or a technical recruiter. Oftentimes, your interviewer will have little or no prior technical experience, especially if you are at the start of the interview process. You may or may not talk with people who you would be working with if you got the job.

You might even have multiple interviews with many different people. Many companies like to get the engineering team involved—they will be the ones working with the new hire—which means that you may have an extra interview with the engineering team before they notify you about whether you have the job. Before you go into an interview, you should try to research who may be interviewing you by using LinkedIn. This can help reduce some of your pre-interview anxiety and allow you to get to know the person you will be talking with.

What to Know

There is not usually any expectations around specific things you should bring to your technical interview. The main things you'll want to bring are standard interview items; such as a pen, a notebook, and a printed copy of your resume. You don't need to bring a laptop to your interview unless the interviewer specifically asks you to—for the most part, you will be answering questions using a whiteboard.

You also need to be aware of the company's dress code. Most tech companies have a casual dress code, but that does not mean you should show up to your interview in jeans and a t-shirt. Rather, you should try to dress above the company's dress code, but not too much. For example, you might want to dress in business casual attire if the company has a very relaxed dress code. If you are unsure about what to wear, research the company online and try to figure out their dress code. You can also ask a recruiter, or if you know anyone who works at the company, you could also ask them directly.

The Stages of a Technical Interview

☎ Phone Screen

Take-Home Task

Technical Interview

There are three main stages in each interview process: the phone screen, the take-home task, and the technical interview. Let's break these down and discuss them individually.

The Phone Screen

A recruiter liked your resume and cover letter, and they are interested in talking with you. Congrats! Now you'll have to participate in a phone screen, where you will talk directly with a recruiter or another member of the company's staff.

The phone screen is all about allowing the recruiter to get to know you better. You will not be expected to answer any highly-technical questions; most will be focused on finding out about your interest in the job, and your soft skills. You don't need to do much preparation for a phone screen, other than thinking about what soft skills you have, and practicing how you can communicate those skills.

You should tailor all of your answers to the company who is interviewing you. If you are asked why you want to work for the company, don't give them a generic answer that could be applied anywhere. Give them a specific answer that makes them feel as if you have thought about the company and how you can contribute to their organization specifically. This personal touch will ensure the interviewer knows you are applying for a job at their company, not just any tech company.

Here are a few tips on how to succeed during the phone screen:

- Be honest. Tell the interviewer about your interests, what projects you would like to work on, and what skills you have. It's better to be honest upfront so you don't get caught off-guard later in the interview process.

- Show, don't tell. If you are asked any questions about your skills or characteristics, show them. If you're passionate, make sure you sound passionate in your interview. If you have a lot of technical experience, tell them to check out your portfolio. If they ask about a project you have worked on, show them how you were involved.

- Also, make sure that you don't assume what the interviewer knows about you. You may be asked to talk briefly about a past project you have worked on, and you shouldn't expect that the interviewer knows the ins and outs of the project. Even if they have read over your portfolio thoroughly, it's unlikely they'll remember every detail about your work. Thus, you should be clear with what you are talking about, and make sure the interviewer is clear on what you are discussing. Regularly stop to ensure your interviewer is still following along, and show an interest in making sure the interviewer understands what you are talking about.

Here are a few other things you can do to set yourself apart from other candidates at this point in the interview process:

Remember that this interview is about testing your coding skills. You should prepare for this interview by doing a few basic practices in whatever language you'll be using in your interview (which the interviewer should tell you). Don't spend too much time preparing for interpersonal questions in this interview—it's mainly about your coding skills.

Do more than the minimum. Sometimes coding challenges—especially take-home ones—will have an extra task that you can complete. To set yourself apart from other candidates, do that challenge as well—doing so showcases you are passionate about the job.

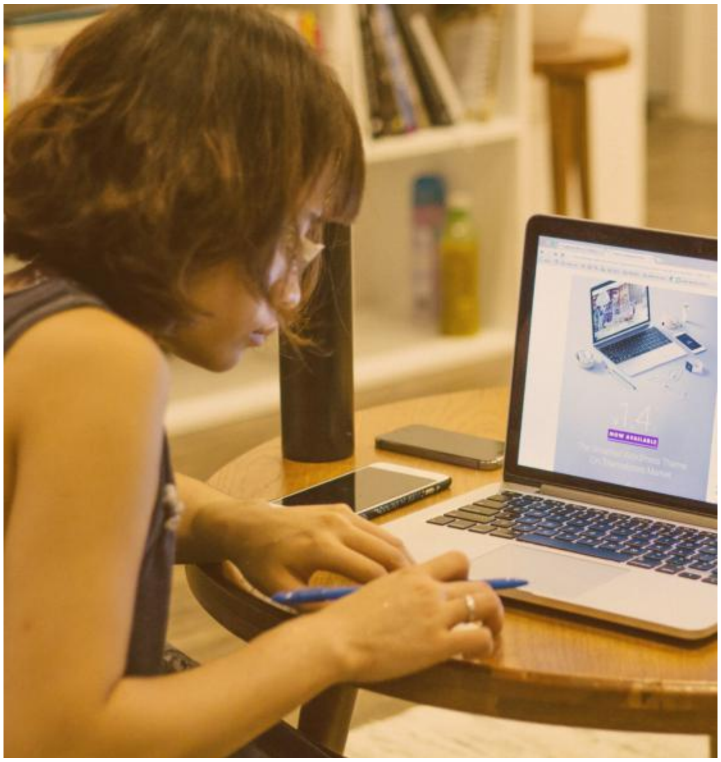
Take-Home Task

If you have successfully completed the phone screen, you may be asked to complete a take-home task. This is often a simple challenge designed to assess your technical skills before you come in for an interview. That said, the technical challenge may still take up a few hours of your time and require your undivided attention. Some companies don't even do take-home challenges, but you should be prepared for one just in case it comes up.

The way a take-home task is conducted will be different depending on the company. Some companies will give you a tool, such as CoderPad, and a programming challenge to solve in real-time. You may do this challenge over a video call and be expected to justify your decisions behind using certain techniques during the interview. Other companies may ask you to complete a challenge within a certain time period—usually a few days.

If you have been given a take-home coding assignment with a time limit, there are a few things you should keep in mind. Firstly, you should take time to plan out what you are going to code before you start working. As we'll discuss later in this guide, writing out a quick plan before you start makes you more likely to cover all the bases and write an efficient program. In addition, you should make sure that you re-read the instructions until you fully understand them. And as you make progress in your program, read the instructions again to ensure you are still working within the parameters of the assignment.

If you are doing a real-time coding challenge, there are also a few things you should keep in mind. You need to be deliberate when doing the coding challenge. Real-time challenges are usually not timed, which means you can feel free to take time when you need it. Before you get started, take some time to think about how you are going to solve the challenge, and write out a short plan. Make sure that you don't rush into things; take it slow, and be careful. Having a plan will help ensure that you stay on-track throughout the real-time challenge, and increases the chance you have a good program to present at the end.



The Technical Interview

Congrats! You've breezed through the phone screening, and you have passed your take-home task. Now comes the technical interview—perhaps the most intimidating component of the whole tech hiring process. You should be as prepared for the technical interview as you can. While the exact format will vary depending on the company size—small companies may use a less formal structure, whereas large companies may have a whiteboard room specifically for the interview—there are a few things you should know about any technical interview.

There are three types of questions you can expect during your technical interview: technical, behavioral, and design-based. Let's explore each type in-depth.

Technical Interview Questions

Technical questions will be focused on a specific technology, language, framework, or a general programming concept; depending on the position for which you are interviewing. For example, if you're interviewing for an iOS developer position, you can expect to be asked questions about Swift, UIKit, and so on.

Before you start the technical interview, you should practice using platforms such as Hackerrank and Codewars to refine your programming skills and practice for your interview. These platforms not only allow you to refine your technical skills in a specific technology, but they can also help you revise the basics such as algorithms and data structures which come up in most interviews.

Technical questions could take two forms. You could either be expected to provide an oral answer, or write about them on a whiteboard. Oral questions will require a well thought out verbal answer, whereas whiteboard questions will usually involve you solving a particular program as if you were coding. You should practice both of these types of questions before you enter the interview room so you know what to expect.

Technical interviews will usually last around an hour each, and most of that time will be spent on technical questions. Before you go into an interview, make sure you have practiced answering questions and challenges for this long period of time.

Behavioral Interview Questions

Companies are not just interested in learning about you as a coder. They'll want to get to know you as a person so they can figure out whether you'll fit in with their culture. Therefore, in a technical interview you can expect to be asked a few behavioral questions such as "Why did you apply for this position?"

Here are a few more examples of behavioral questions you may encounter in your technical interview:

- Talk about a time you failed and how you handled that failure.
- Which of our company's values is your greatest strength?
- Tell us about a time where you solved a conflict at work.
- Talk about your approach to solving difficult problems.
- Give an example of how you set goals, and how you achieve those goals.
- Tell us about a time you went above and beyond expectations.
- What do you do if you disagree with your manager?
- Who has influenced you the most in your career?
- How do you stay organized?
- Talk about a time you had to make a decision with a lot of ambiguity.

When you are answering behavioral questions, there are a few tips you should keep in mind:

- Be specific: Your answers need to be as specific as possible. Include statistics, context, technologies, examples, and anything else you think is relevant. The more specific you are, the easier it will be for an interviewer to get to know you.

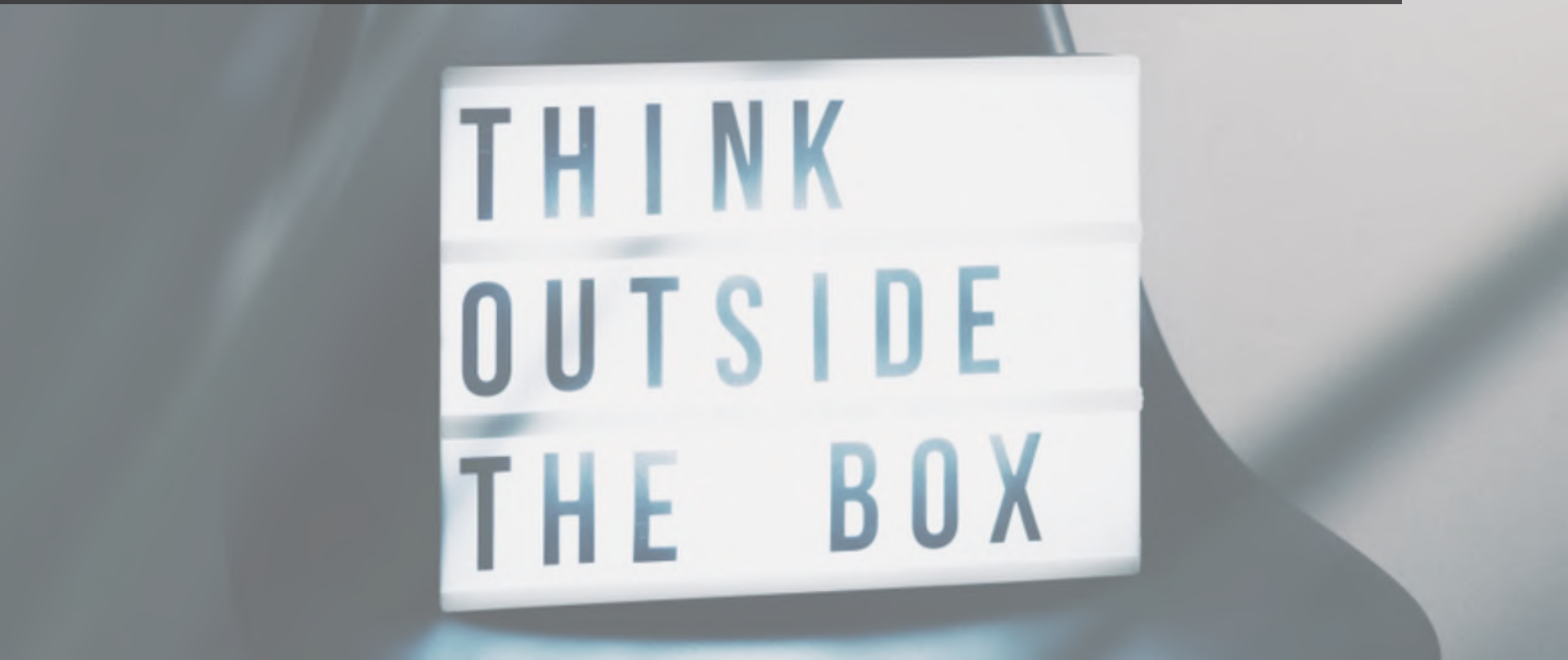
- Avoid negative language: Talk about the great work you have done. Even when you're talking about a challenge you have encountered, try to use positive language. Avoid words which sound like you have failed such as "did not meet expectations" and instead focus on what you would do differently next time.

- Stay focused on the question: Make sure your answers pertain to the exact question you have been asked. Also, try to make sure your answers are concise—there will be a lot of questions the interviewer will want to get through in a short period of time.

Design Interview Questions

Design questions are very similar to technical questions and will relate to your technical skills. However, these questions are often more logical and require big-picture thinking. For example, if you were applying for a system administrator job, you may be asked about how you would structure a server cluster. Or, in a back-end web developer interview, you may be asked about how you would structure an API.

The goal of these questions isn't to have you draw up a perfect diagram of a system structure. Rather, it is to get an insight into your way of thinking about systems. These questions give you an opportunity to show how you would respond to bigger problems and how you would go about solving them. When you're being asked design questions, you should ask as many questions as you think is necessary to ensure that you know everything you need to answer the question effectively.



Tips for a Successful Technical Interview

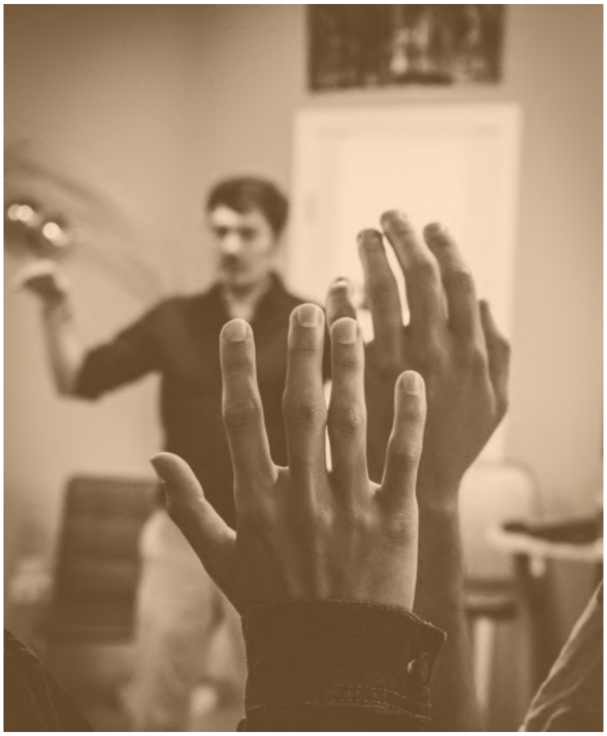
Here are a few tips on how to have a successful technical interview and land your dream job:

1. Speak Clearly

You need to make sure you are being clear and specific when answering questions. If you get distracted or trail off in your answer, the interviewer may leave confused. Although you may think a technical interview is all about coding, interviewers are also evaluating your social skills. Can you communicate well? Can you work well under pressure? And so on. Make sure you speak clearly while giving your answers, and if an interviewer looks confused, feel free to elaborate and break your answer down.

2. Think Out Loud

When you're coding at home, you don't have to talk out loud because you are the only one who needs to understand what you are doing. But when you are in a technical interview environment, it's important that you talk through your code so the interviewer knows what your thought processes are. Talk out loud about why you're using a certain approach to solve a problem. Remember, the interviewer is not only evaluating your final answer. They are also thinking about the route you took to get there and how you work: Can the candidate talk about their thoughts while writing out code? Can they provide clear responses when justifying their coding decisions?



3. Make Sure You Understand the Question

Before you start writing code, take some time to think about the challenge. If you need any clarifications, ask the interviewer to repeat the question or clarify something. It may sound like you are doing the right thing by getting to work immediately, but your brain needs time to think about the problem before you can get started. If you ask questions, you can make sure you know exactly what is expected of you, and you can show the interviewer that you value clarity.

4. Don’t Panic

If you get stuck, don’t panic. Instead, you should think about your ideas out loud, and try out new ideas if what you are doing isn’t working. If you are unsure about the answer to a question, don’t immediately start worrying—that’ll make it more difficult for your mind to come up with an answer.

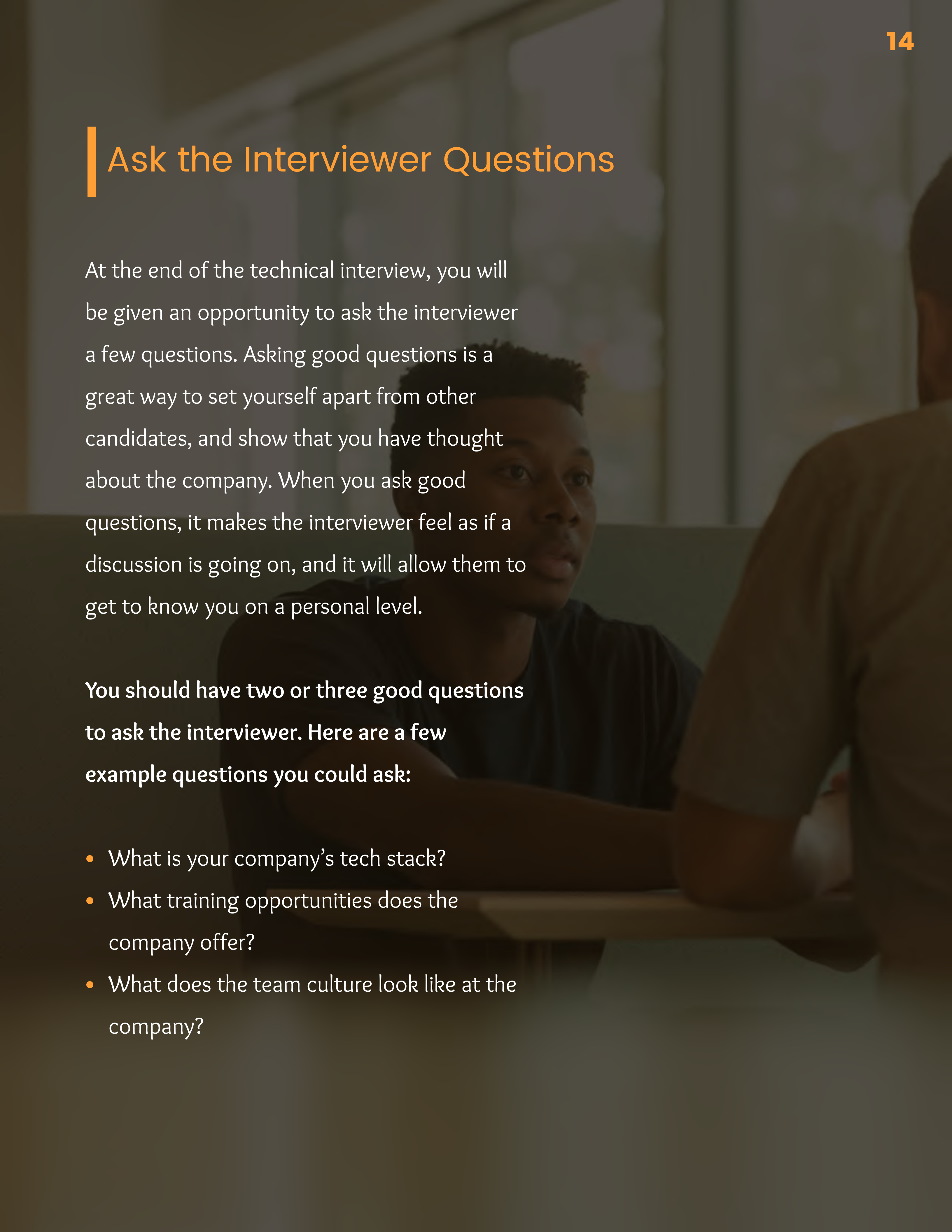
You should instead take a moment to think about potential answers. This is harder advice to put into practice than it sounds, but the one thing you should remember is that the technical interview is your chance to showcase your potential.

It’s not the end of the world if you get stuck. If you really don’t know the answer to a question, tell the interviewer that. And if you can’t come up with a complete answer, show your working anyway on the whiteboard. You may get some credit for trying a problem, especially if it was intentionally complex.

5. Keep Your Solutions Clear

In addition to speaking clearly, you should also keep your solutions to whiteboard questions clear and simple. You are not being rewarded based on the number of lines of code you write. To the contrary, the more efficient you can make a program, the more likely you are to advance to the next stage of the hiring process. When you are designing your solutions, you should also make sure your variable names are clear. Don’t use names only you understand—make them clear and ensure their purpose is stated in the variable name.

The more confused an interviewer is when they are reviewing your code, the less likely they are to reach out to advance the hiring process. And feel free to add comments if you think they will help you think through a problem, or provide clarity for the interviewer when they are reviewing your code.



Ask the Interviewer Questions

At the end of the technical interview, you will be given an opportunity to ask the interviewer a few questions. Asking good questions is a great way to set yourself apart from other candidates, and show that you have thought about the company. When you ask good questions, it makes the interviewer feel as if a discussion is going on, and it will allow them to get to know you on a personal level.

You should have two or three good questions to ask the interviewer. Here are a few example questions you could ask:

- What is your company's tech stack?
- What training opportunities does the company offer?
- What does the team culture look like at the company?

How to Prepare for A Technical Interview

A technical interview is a daunting task. There's a lot that you will need to prepare for; whether it be working on a whiteboard, or answering the types of questions that will come up in the interview. Preparation will help you succeed in the interview and will, in turn, increase the chances that you get the job.

Let's explore a few ways that you can prepare for your technical interview to set yourself up for success:

➔ Study the Basics of Computer Science.

Most technical positions—data scientist, web developer, Android developer—require a good knowledge of CS fundamentals, such as data structures and algorithms. Make sure you are familiar with the basics so that if any of these questions come up, you'll be prepared to answer them. Here are a few common CS concepts you should be familiar with:

- Arrays
- Data structures
- Trees
- Sorting algorithms
- Hash tables

➔ Do Dry-Runs

There is no better way to practice for a technical interview than to do a mock interview with real questions that you can expect to come up during your real interview.

➔ Think About Past Projects

During the interview, you may be asked to talk about a project you have worked on in the past. You should make sure that before you start your interview, you have a few specific projects in mind that you can talk about in-depth.

These projects should be relevant to the position you are applying for, and demonstrate your hard and soft skills in action. For example, you may talk about a time when you built a challenging full-stack web application and discuss how you overcame the major challenges associated with the project.

Having some past projects in mind will allow you to answer these types of questions quickly and effectively. If you don't have a track record of working on professional projects, build a portfolio of side projects which showcase your skills.

What to Avoid in a Technical Interview

There are a few pitfalls to avoid during your interview. Let's break these down so you know what not to do.

1. Failing to Prepare

You need to spend as much time as possible preparing for your interview, because you'd be surprised how many people don't: Preparing adequately can easily set you ahead of everyone else. Be ready. Don't think that you will just figure it out in the room. You know the old adage: if you fail to prepare, you are preparing to fail!

2. Being Impersonal

During an interview, it can be tempting to solely focus on your technical skills. But the interviewer wants to get to know you as a person too. Therefore, you need to act like a human in your interview—you need to make conversation, and be yourself. Laugh at any jokes if it is appropriate. When you walk into the interview, be polite and personable. If the interviewer leaves thinking that you are great at working with others, they will be more likely to reach out to you after the interview and offer you the job.

3. Rushing to an Answer

You only have a limited amount of time in which to answer questions. Thus, it can be easy to rush to an answer so you can make your way through the interview quicker.

However, no matter how interested you are tackling a problem or getting through the interview quickly, you need to take a few minutes before answering a question to stop and think. Make a plan in your head—or on the whiteboard—about how you are going to go about answering the question. This will allow you to make sure you have a clear sense of direction when working, and will reduce the chance that you make basic mistakes in your interview.

Interviewing is a skill and it can be improved with practice. Mock interviews allow you to simulate the interview environment—being asked questions by someone else, being bound by a one-hour or so time limit, et cetera—and prepare you mentally for what you can expect. Doing a dry-run may also make you feel more confident about going into the interview room.

4. Refine Your Elevator Pitch

Like founders pitching their company, you should have a short elevator pitch which features your job history, past projects, and anything else that might be relevant. This pitch should be around one minute long and tell the interviewer the main things they need to know about you. By preparing an elevator pitch, you'll be able to effectively answer the dreaded "tell me about yourself," question, which often feels unnatural to be asked. Think about your main accomplishments, and figure out how you can articulate them in a short and concise manner.

5. Know the Company

Do some research on the company. Find out what their long-term ambitions are, and who they work with. What are their products? Are they a direct-to-consumer or business-to-business company? Researching what the company does and how they work will prepare you for the end of the interview when you can showcase your interest in the company by asking pointed questions.

You can find out more information about the company you are interviewing at on platforms such as LinkedIn or Glassdoor. These sites often feature reviews of larger companies' cultures and hiring processes.

6. Understand the Basics of Your Particular Field

You'd be surprised how often interviewers ask basic questions. You may think that a technical interview will be filled with complex challenges, but some of the questions are actually simple. But it is often because they are simple that people slip up. You need to make sure you know the basics of whatever technologies you claim to know, just in case they ask you any questions.

Go over the basics, and make sure you know them inside out. Because you may also be given open-ended questions which are designed to allow the interviewer to assess your grasp on the fundamentals. Those types of questions can also allow the interviewer to understand your technical mindset in more depth, as there is no specific right answer.

7. Learn How to Manage Your Time

Your interview will only last an hour or so, and you should know how to use that time effectively. Set timers when you're doing practice questions. This will help you prepare for the actual interview where the interviewer will have a list of things they want to cover in a short period of time.

What to Do After a Technical Interview

Well done on completing the technical interview! The most difficult part of the hiring process is now over. Now all you have to do is wait to hear back from the company about next steps. However, there are a few things you should keep in mind with regard to what to do after your technical interview.

Firstly, you should remember that this job is not the only one out there. Indeed, it can feel bad to be rejected, but you should remember at all times that if you are rejected, there are always other jobs you can apply for.

Don’t tell yourself that you are definitely going to get the job, as that can make you feel worse in the case that you don’t. And even if you don’t get the job, congratulate yourself on advancing so far through the interview process (you got there for a reason), and think about the interview as practice for the future.

You may also want to consider sending a thank you email to the hiring manager and other people involved with the technical interview. This shows that you appreciate the time the company has invested in screening you, and also that you value the company. Sending a thank you email will also make you stand out from everyone else—most people will not go to this effort.

Conclusion

With all of the tips we have covered on how to succeed in a technical interview, you're ready to walk into the interview room and get started. The technical interview may seem intimidating, but you should think of it as your best opportunity to show off your skills to someone new.

In the end, keep practicing, revising, and learning. The more you practice, the better your chances are of getting your dream job!

About Us

Career Karma helps people interested in breaking into careers in technology find the best job training for them, starting with coding bootcamps.

With the free Career Karma app, you can discover peers, coaches, and mentors to help accelerate your career in technology. Career Karma also publishes comprehensive insights into how to pick the best coding bootcamp and break into a career in technology.

James Gallagher is a researcher at Career Karma, where he focuses on student debt, higher education reform, and how Income Share Agreements can make higher education more affordable. James analyzes market trends influencing the higher education landscape, and his research involves studying Income Share Agreement policy, accountability in higher education, and various proposals for mitigating the rise of student debt.

Next Steps?

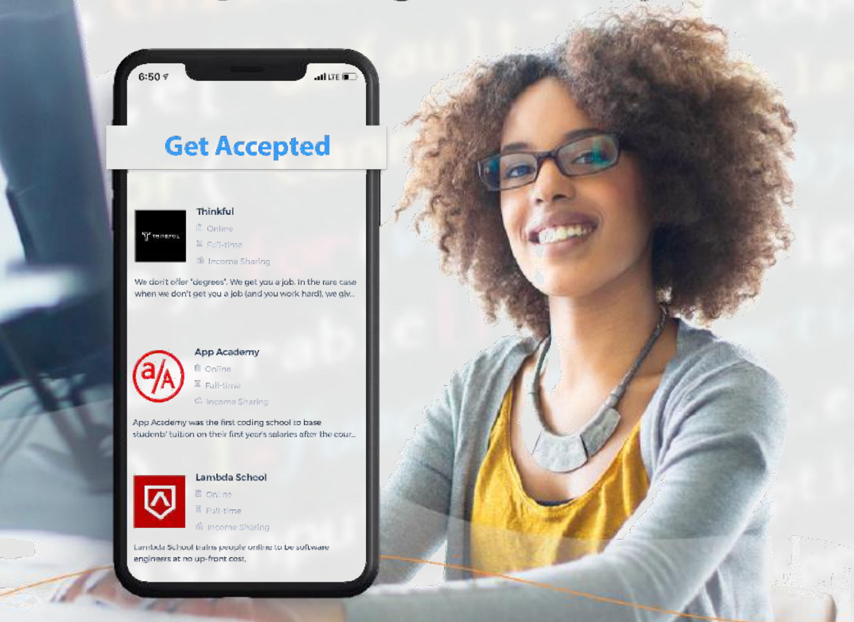
Get more useful information and advice on how to launch a tech career with the **free** Career Karma app!

Download on the App Store

GET IT ON Google Play

Download today and join a community of peers, coaches, and mentors who will support you in your journey to a new career in tech!

Get offers from top coding bootcamps



6:50 LTE

Get Accepted

Thinkful
Online
Full-time
Income Sharing

We don't offer "degrees". We get you a job. In the rare case when we don't get you a job (and you work hard), we giv...

App Academy
Online
Full-time
Income Sharing

App Academy was the first coding school to base students' tuition on their first year's salaries after the cour...

Lambda School
Online
Full-time
Income Sharing

Lambda School trains people online to be software engineers at no up-front cost.

CAREER KARMA.