18 [Effective Interviewing](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/17346)

18.1 [The Software Engineering Interview Process](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/17347)

1. A Beginner's Guide To The Technical Interview  
   <https://learntocodewith.me/posts/technical-interview/>
2. Software Engineering Mock Interview Guide   
   18.1-1 문서 참고
3. 10 Essential Software Engineering Interview Questions   
   [https://www.toptal.com/software/interview-question](https://www.toptal.com/software/interview-questions)s
4. A Developer's Guide To Interviewing Optional  
   <https://www.freecodecamp.org/news/how-to-interview-as-a-developer-candidate-b666734f12dd/>
5. What To Ask Your Interviewer

18.1-2 문서 참고

18.2 [The Hiring Process](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/58878)

In the tech industry, the hiring process refers to the steps that companies take to identify, recruit, and hire technical talent. Due to the technical nature of the industry, the hiring process in tech can be more complex than in other fields. The hiring process starts out with a recruiter screening. The recruiter will ask you basic HR questions that include “Tell me about yourself,” visa status, and salary requirements. You will then move to a Hiring Manager screen, where the hiring manager will look for high-level technical capabilities and culture fit. You will then have a technical interview. The company will then decide to move to hire.

1. Interviewing Stages Expectations (1 pages)   
   18.2 문서 참고

18.3 [How to Prepare for Your Interview](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/58880)

To prepare for an upcoming interview, you must review the company website to ensure that you understand the company and its goals and prepare relevant and insightful answers to interview questions.

1. Conducting Company Research (1 pages)   
   18.3-1 문서 참고
2. The STAR Interviewing Response Method  
   [STAR interview method in 2022 (UPDATED)](https://youtu.be/h7IlUFQRsJw)

18.4 [Take Home Challenges](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/17359)

These take home challenges will replicate the kind of take home challenges you might get during a coding interview. When interviewing, you may receive a project to complete within a limited timeframe. The project should be challenging but doable, and your potential employer will be looking at your solution, the way you format and comment your code, and if you include tests. We've given you a few practice take home challenges to try out.

1

1. Take Home Challenge 1  
   <https://github.com/rithmschool/yodlr-take-home>
2. Take Home Challenge 2  
   <https://github.com/rithmschool/outcomes_curriculum/tree/master/challenges>

18.5 [More On Whiteboarding](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/17365)

In this subunit, you'll continue learning about whiteboarding. In the last unit, Colt discussed tactics to succeed at whiteboarding interviews. There's a lot of information out there regarding whiteboarding, so we've got some more resources to help you ace this portion of your interview.

1. Knock Out That Whiteboarding Interview  
   [https://skillcrush.com/blog/rock-your-next-whiteboard-test](https://skillcrush.com/blog/rock-your-next-whiteboard-test/)
2. The Best Whiteboard Interview Advice I Ever Received  
   <https://hackernoon.com/the-best-whiteboard-interview-advice-i-ever-received-3ebbfa72e4a>

18.6 [Coding Interview](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/17368)

1. What Do Coding Interviews Look Like?

<https://www.linkedin.com/learning/get-ready-for-your-coding-interview/>

1. How To Ace The Coding Interview   
   <https://docs.google.com/document/d/1Gx6C1kuOmoQ0gz_2r9T3RzCjh2ZFkaxTdP1F-OZ-0pM/edit?tab=t.0#heading=h.7o6kkljy676u>

18.8 [Interview Follow-Up and Your Mock Behavioral Interview](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/58883)

18.8 문서 참고

18.9 [Wrap Up](https://www.springboard.com/workshops/software-engineering-career-track-f2c/learn#/curriculum/17376)

In this unit we addressed the different types of interviews you will want to prepare for as part of your software engineering job search, including a behavioral interview, a portfolio presentation, a whiteboarding technical interview, and as part of those, a phone screen and take home challenge as well. You should have booked your mock interviews to practice each of these components. Your career coach can be a great resource to make sure that you are well prepared for your interviews moving forward.