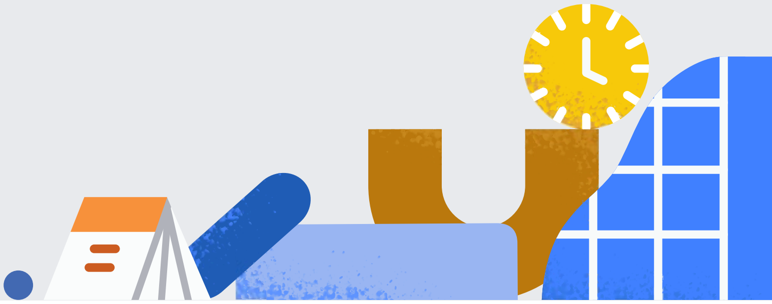
FACEBOOK

FRONT-END ENGINEER

Onsite Interview Guide



What You'll Find in This Guide

Interview overview

Coding rounds prep

Design / architecture round prep

Career and coding conversation prep

Appendix / resources

Welcome to your prep guide for your front-end engineer (FEE) onsite interview at the Facebook company. Our front-end engineers put together this guide so you know what to expect and how to prepare.

Interview overview

What will your onsite interview be like?

Your onsite interview will include several back-to-back interview rounds with our engineers. Here's a high-level view of what to expect.

Content

You can expect to cover the following areas in your interviews: two coding sessions, one design / architecture session, and one behavior / coding session with an emphasis on culture, growth opportunities, and impact.

The primary technical focus throughout our interview process will be JavaScript, HTML, and CSS. But know that at Facebook we consider our FEEs to be software engineers first, and web specialists second. This means there may will be discussion of general computer science concepts like algorithms, design patterns, data structures, etc. We'll try to keep it practical and focus on things you might encounter when building apps though.

Agenda

Your conversations with our engineers will be divided into the following time blocks:

- **Two Coding Rounds:** 45 minutes
- **One Design / Architecture Round:** 45 minutes
- **Lunch With an Engineer:** 45 minutes
- **One Career Conversation:** 45 minutes

How to Avoid Pitfalls

A common mistake is to look at interview problems, recognize them, and sort of understand them, but not to be able to actually code them. So, just to be totally clear, you should prepare by practicing writing code by hand, without a computer. If you prepare by solving interview-style questions in a timed way, you'll increase the chance that your coding ability will come through during the interview process.

Coding rounds prep

What to expect

You'll have two 45-minute interviews that focus very heavily on coding. They'll be similar to your initial interview, but count on these being a bit more challenging. The engineers will be looking for accurate, bug-free, efficient, and well-thought-out code via whiteboard coding questions. The questions will involve data structures and algorithms, but with a practical focus on what you might encounter building user interfaces at Facebook. The engineers will want to hear your thought process throughout, so be sure to provide a narrative as you go through the code.

Not all interviewers follow the exact same breakdown, but they typically include the following:

- **Introductions:** The first couple of minutes will be an introduction, and possibly brief questions about your background and your specific knowledge.
- **Coding:** The next 30 – 45 minutes will be one or more coding problems.
- **Ask Us Anything:** We try to reserve the final few minutes for your questions for the interviewer. This part gives you a chance to learn more about Facebook from someone in engineering, and gives your interviewer a chance to learn more about your interests.

How to prep

As you begin preparing, please **watch this video** (password: fbprep), which gives a good example of what Facebook coding interviews are like. While the code in this example is Python, we use many different coding languages at Facebook. The front-end interview will focus on JavaScript, so you'll want to practice in JavaScript specifically.

You should be able to whiteboard solutions from simple to medium difficulty, programming interview questions in under 15 minutes.

To prepare, it's not enough to read through sample questions and recognize the concepts. You need to practice the actual writing of the code without a computer, simulating a timed interview environment. When you have a solution, you should look at it and see if it's something that you'd approve if it were submitted to you as a proposed part of your codebase. Make sure that it's correct, that you have taken into account the edge cases, that it's efficient, and it clearly reflects the ideas that you're trying to express in your code. In addition to reviewing the CS fundamentals, these tips may be helpful:

- **Understand the problem** you have to solve. It's OK to ask for clarifications or to talk through the problem.
- Problems often have more than one solution, so be ready to **discuss constraints and tradeoffs**.
- It's not imperative that you've memorized tons of DOM, JavaScript and browser APIs, but we'll be looking for **familiarity and experience.**
- Get comfortable with JavaScript. You need to have a clear understanding of variables, functions, objects, arrays, asynchronous control flow, and closures. Check out **https://javascript.info/** to refresh your knowledge.
- Think about different algorithms and algorithmic techniques (iteration, sorting, divide-and- conquer, memoization, recursion).

- Think about data structures, particularly the ones used most often (Array, Stack / Queue, Object Hash, Tree, Set, Map, etc). While the front-end interview doesn't focus on building them from scratch, we expect that you can improve UI performance by choosing the appropriate data structures.
- **Modifying the problem** or thinking about it in smaller pieces may be helpful.
- **Practice coding on a whiteboard**.
- Practice resources: **Topcoder, GeeksQuiz, CareerCup**, Cracking the Coding Interview, etc. (reference your initial screen resource guide for more). Keep in mind that the Facebook front-end interview is focused on the concepts relevant to UI engineering so practicing CS-heavy questions like balancing red-black trees will not be very relevant.
- Check out a **mock coding interview** at Facebook (password: 2018prep).

Design / architecture round prep

What to expect

The design interview is 45 minutes and it doesn't include coding. Instead, you'll spend the interview talking and using the whiteboard as a visual aid (think: box and arrows diagrams, function signatures or API specifications). As with all interviews, the interviewer will typically save the last five minutes for your questions.

What we ask

The purpose of the interview is to assess your ability to solve a non-trivial engineering design problem. The interviewer will be trying to determine if you can architect a solution to a higher-level problem that requires connecting multiple concepts. Typically, we're looking to see that you have a good understanding of some of the concepts we care about – like browsers, the DOM, performance, API design, etc. To that end, your interviewer will ask you a very broad design problem and evaluate your solution. We try to match candidates to engineers with related expertise.

How to prep

Go into it knowing this is the hardest interview to study for. Use the prep tips here, and in the resources listed in the appendix.

- **Improve upon a design:** Think about and review the complex systems you've already designed. What would you change about your approach if you did it all over again? What worked well?
- **Design from the ground up:** Think about how you'd design a system that Facebook (or another large tech company) already has. It's a good exercise to think through the complicated, high-scale systems that you already use every day. How would you design it from scratch?
- **Do some research:** Read engineering blogs about approaches that have worked for big companies along the way. Read about the false-starts too!

How to Avoid Pitfalls

- It's not an interrogation! Don't forget that you're driving the discussion.
- The interviewer might ask questions along the way, but this is your time to really own the conversation, so put yourself in the driver's seat. This isn't a typical "answer the questions" interview.
- Do all your prep by whiteboarding, diagramming and illustrating concepts.
- Prep by deep-diving and picking one or two facets of the problem, and be ready to show technical breadth and depth in these areas.

- **Start with requirements:** Your interviewer might ask: "How would you architect the front-end for a messaging system?" Obviously this question is extremely vague. Where do you even start? You could start with some requirements:
    - How many users are we talking about?
    - What should the experience be while you're waiting for confirmation?
    - How will you show error states?
    - What are the latency requirements for sender–receiver message delivery?
    - How are you going to store messages?
    - What kind of features are we going to need to support?
    - What operations does this data store need to support?
    - How do you push new messages to clients? Do you push at all, or rely on a pull-based model?
- **Ask us anything:** The last few minutes are for your questions. This is a great opportunity to get an insider's perspective directly from a Facebook engineer.
- **Design tutorial:** This is a good **online tutorial** about design questions.

Design / architecture interview tips for the day

- Explore the tradeoff: Make sure you bring up competing solutions and speak to all their major tradeoffs. Demonstrate that you can make good decisions about how to balance the various tradeoffs.
- Drive the discussion with the interviewer: You'll be steering the conversation, and we expect you to understand the problem by asking clarifying questions.
- Treat the interview as a conversation: Be sure to ask clarifying questions and make sure you drive towards a good, working solution.

How to Avoid Pitfalls

- Familiarize yourself with our **5 Core Values** (Move Fast, Be Bold, Focus on Impact, Be Open, and Build Social Value). This is how we work together to make the world more open and connected. We look for people who believe in these values and practice them daily.

Career conversation prep

What to expect

You'll meet with our engineers to talk about about your career. They'll ask about you, your history, your résumé, and your motivation. Yes, the "you" part is behavioral in the sense that it helps the team assess whether you'll thrive in Facebook's peer-to-peer, minimal-process, collaborative engineering organization.

How to prep

- **Know yourself:** Take the time to review your own résumé as your interviewer will almost certainly ask about key events in your work history.
- **Be honest:** Not every project is a runaway success. We may not always interact perfectly with our peers. The most important thing is how we learn, improve and grow from our past experiences.

How to Avoid Pitfalls

- Be yourself! Be open and honest about your successes and failures.
- Be humble and focus on teamwork, leadership, and mentorship qualities.
- This **video** by Jackson Gabbard (former Facebook engineer) breaks down the goals, components, and best practices of behavioral interviews.

- There are several behaviors that the interviewer will assess during this interview: conflict resolution, empathy, motivation, initiative, perseverance, self-improvement, communication, and dealing with ambiguity.
  - Prepare one or two short examples from your career or past experience that speak to each of these behaviors.
- Use the SOAR method to mentally organize your thoughts. This will provoke a well-thought-out and chronological order of events. Easy to describe, easy to follow.
  - S—One or two sentences about the **SITUATION**—What happened?
  - O—Describe the **OBSTACLES** you encountered: What made this difficult?
  - A—**ACTIONS** you took to overcome the obstacles and complete your objective: Describe the actions you took to overcome the obstacles and achieve your goal.
  - R—The tangible / quantifiable **RESULTS** of the situation: Explain the tangible or quantifiable outcome—How did it help the team / company?
- **Have concrete examples or anecdotes:** Support each question with examples. Some typical behavioral interview questions are:
  - What were some of the best things you've built?
  - What are you proud of?
  - Tell me about a project that failed and what you learned from the experience.
  - What were some excellent collaborations you've had?
  - Tell me about a time when you advocated for and pushed your own ideas forward despite opposition?
  - How do you deal with conflict?
  - How do you like to give and receive feedback?
  - What is the best advice you have received from a peer or mentor?
  - Why Facebook?

Appendix / resources

Links to exercises, information, and guides to help you prepare

Our team collected some helpful resources with content and activities for your onsite interview. Take a look through the list as you prepare.

About Facebook

- **5 Core Values**
- Learn about life at Facebook on Instagram via **@facebooklife**
- Join our **Facebook Careers Talent Community Page** for the latest updates.
- **Facebook Engineering Bootcamp**
- **Facebook Newsroom**

Facebook Engineering and Open Source

- **Facebook Engineering**
- **Facebook Open Source**

Video Resources

These resources contain basic procedural concepts. Use the guide for the language you plan to use during the interview.

- **Intro to Behavioral Interviews**: This video by Jackson Gabbard (former Facebook engineer) breaks down the goals, components, and best practices of behavioral interviews.
- This is a good **online tutorial** about design questions.

Coding Practice

- **Mock Coding Interview at Facebook** (password: 2018prep)
- **CareerCup**

Javascript resources

- **Learn JavaScript** (MDN)
- **A Re-introduction to JavaScript** (MDN)
- **The Modern JavaScript Tutorial**
- **JavaScript Garden**
- **Eloquent JavaScript**
- **Understanding ECMAScript 6**
- **Learning Advanced JavaScript**
- **5 Typical JavaScript Interview Exercises**

Thanks for taking the time to review this guide and good luck in the interview - you'll do great!