

A good interview experience can make or break a candidate's decision to take the job, as well as form their opinion of your employer brand. We've broken down the interview experience into three stages, all of which are of equal importance.

#### **Pre-interview**



#### Better prepare them ahead of time

35% of developers say they want to be better prepared for the interview, including being informed of who they will be speaking with. Send out an email with the basic details to the candidate a few days before their scheduled interview to ensure they are well prepared. This also gives them time to ask any pertinent questions.



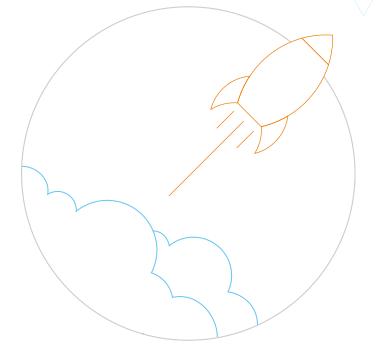
#### Be flexible about interview scheduling

20% of developers say this is important to them when considering a job interview. Getting time away from a job during the day can be difficult (not to mention awkward), so show some empathy.



#### Sort out expectations up front

This is especially true regarding salary, which can make or break a job offer for a candidate. If the expectations of the job are not a good fit, don't waste any further time. Everyone will benefit from the honesty.













# **During the interview**



# Be on time and organized

Being late or disorganized is a terrible first impression and is likely to scare off good candidates.



#### Introduce them to the team

47% of developers want to be introduced to the team during the interview process. Programmers will pay very close attention to the people they meet, focusing on things like "Are they nice?" and "Are they smart?" Additionally, 19% of developers say they want to be introduced to the boss. This isn't necessarily feasible or relevant in every situation, so think about if it works for you and go from there.



# Give them a small tour of their workspace

Obviously this only applies to in-person interviews. 37% of developers would like the opportunity to see the workspace during their interview. "The bottom line is that programmers are more likely to take the job that offers them a private office"- Joel Spolsky, Smart & Gets Things Done. Cubicles and shared spaces can be awkward and allow for easy distractions, and private offices make software developers more productive. When candidates come in for their interview, they will try to picture themselves working there. They'll pay special attention to the hardware setup (monitors, types of computers, etc.) and if there are comfy chairs, so keep that in mind. It's understood that not every office can realistically have private offices or expensive chairs, but try your best to accommodate to the developer, such as providing a distraction-free space or noise-cancelling headphones.











# Try to have multiple people interview each candidate

This includes a few employees who would be peers of that candidate (other programmers or members of the Product team, for example). Asking the candidate to spend hours with non-technical people will likely bore them, and feel like bureaucracy, so be careful. Another alternative is to have a team lunch where everyone can meet and talk briefly.

#### After the interview



Write down immediate feedback on the candidate while it's fresh



# Communicate the decision with the candidate as soon as you know

You don't want them taking another job in the meantime. You also don't want to just stop communicating if they didn't get the job – it leaves the candidate stressed and feeling uncertain, so show some empathy.

"Many developers say one of their biggest complain is applying and interviewing into a 'black hole."

Matt Sherman, Engineering Manager at Stack Overflow

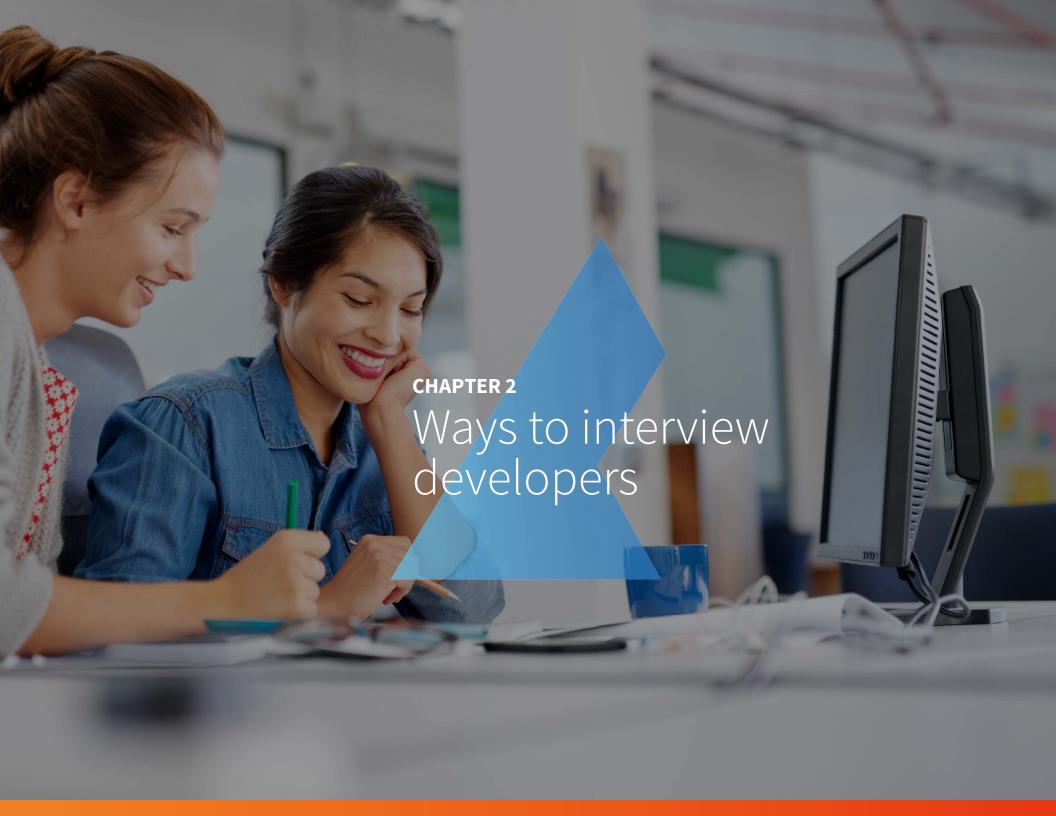












Everyone has a different theory on the best way to interview a developer. Here are a few of the more common methods.



# Have them participate in live coding tests

- 34% of developers surveyed want to see more live code in the interview process.
- Oftentimes these live coding tests are referred to as "whiteboard programming", which is just what it sounds like—writing code on a whiteboard. This type of test helps to evaluate how quickly and clearly a candidate can articulate their thought process.



# Put them through a phone screen

Phone screens are used to make sure that the company isn't wasting time and money on someone who isn't smart.

- Cost effective.
- Eliminate about half of the people who looked good on paper.
- Are very fair (you can't see them and therefore focus more on what they're saying than on other external factors like appearance or nervousness).



#### Use video interviews

- Save time, are cost-effective (no travel), convenient, and recordable.
- Types of software used to conduct video interviews:





(codassium);

**Interview Zen** 



David Haney, Engineering Manager at Stack Overflow



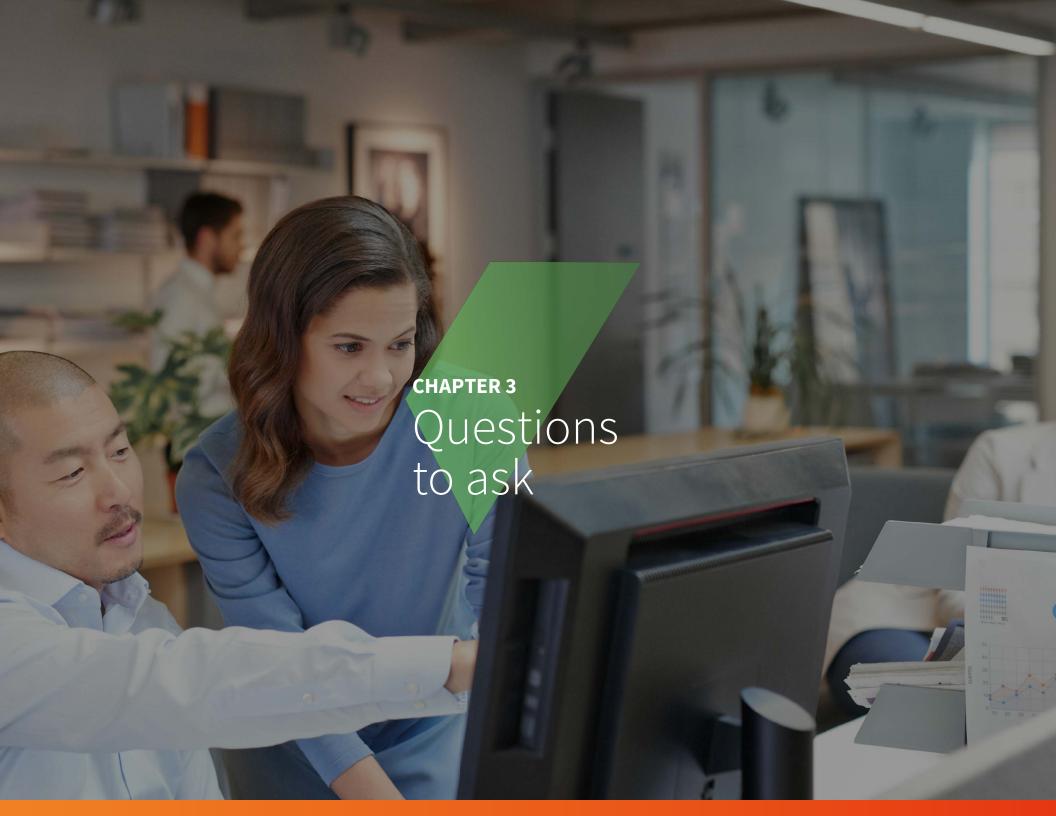












Joel Spolsky, CEO of Stack Overflow, splits up his programming interviews into a few separate parts. To follow his framework, focus on the questions below.

# Introduction



#### Puts the candidate at ease

Allows the interviewer to talk a little about themselves briefly.



# Ask about a recent project the candidate worked on

- Which technologies did you use?
- What role did you play in the process?
- How did you handle challenges during this project?



# Ask them to describe their career history

From these questions, you should be able to see their passion and leadership qualities. They should be able to explain themselves well.











# New heading?



#### Easy programming question

Use a whiteboard. Explain beforehand that it's hard to write code without an editor and that will be taken into consideration. Some examples from Joel:

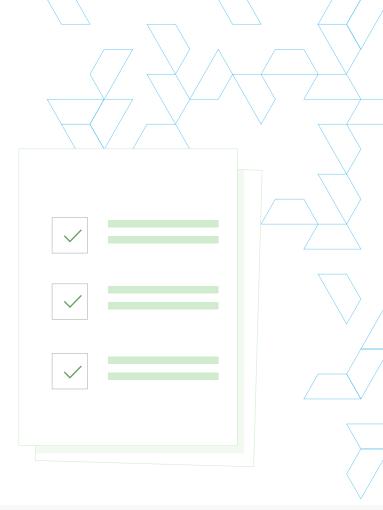
- Write a function that determines if a string starts with an upper-case letter A-Z.
- Write a function that determines the area of a circle given the radius.
- Add up all the values in an array.



#### Algorithmic/recursion question

The goal of this type of question is to have a conversation with the candidate. You could potentially ask the following questions as they solve the problem:

- Why did you do it that way?
- What are the performance characteristics of your algorithm?
- What did you forget?
- Where's your bug? (This is inevitable, so asking them to find it is a legitimate question).





# Other questions to ask

- What programming blogs, websites, or books do you read frequently?
- Tabs or spaces? (Sure to start a debate).
- What cool or exciting things have you learned lately?
- Anything about current events in the tech industry.
- What are the most important things to you in a programming job?

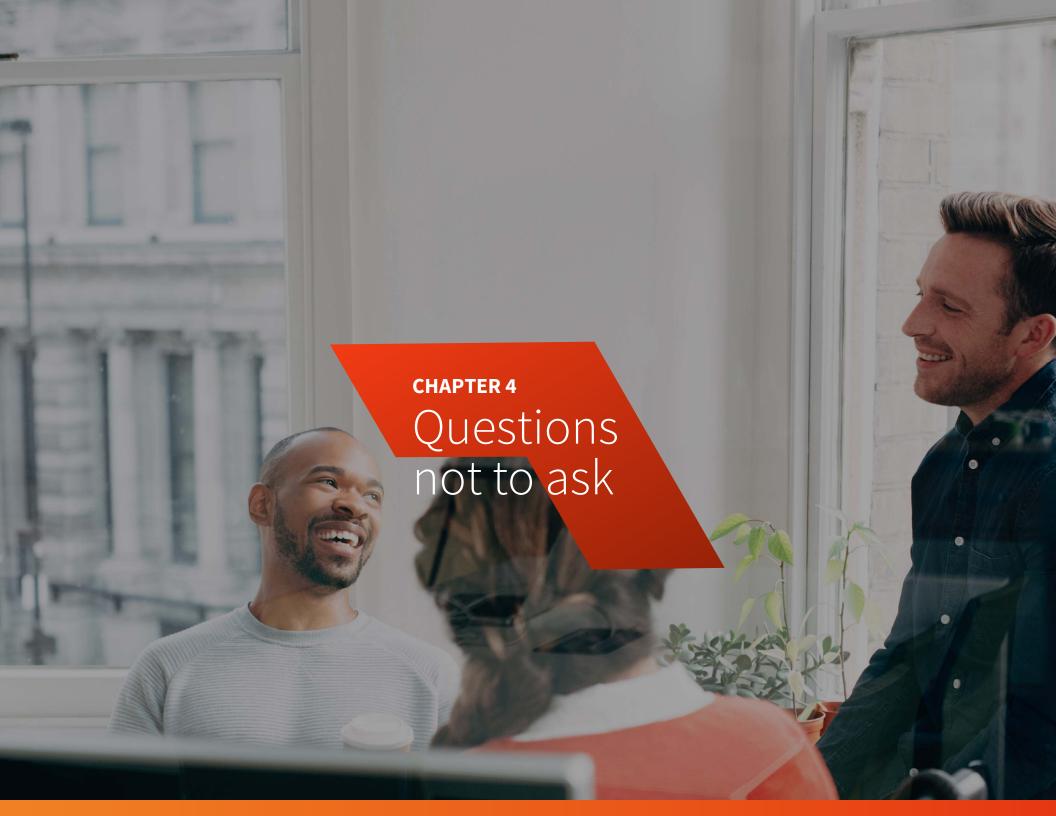












CHAPTER 4
QUESTIONS NOT TO ASK



#### **Brainteasers**

28% of developers say they wanted to hear fewer brainteasers in the interview process. These types of questions are essentially useless. They mainly evaluate whether the candidate has heard the odd questions before (or Googled the answer ahead of time in anticipation).



#### Quiz show questions

Asking what the "var" keyword does in C# or what "final" means in Java is mostly useless as it is transient knowledge. Any developer can Google it and figure it out in minutes on the job as needed.



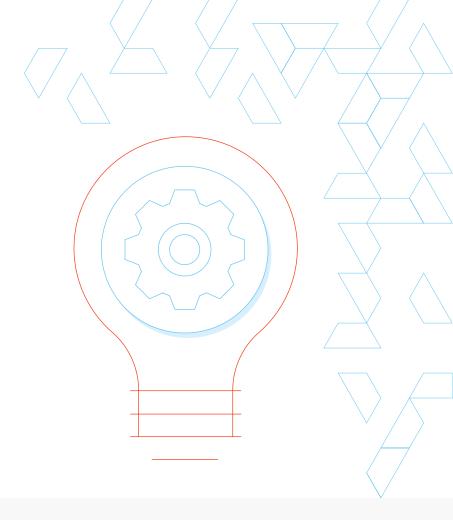
# Illegal questions

Any questions related to race, religion, gender, age, sexual orientation, etc.



# Lifestyle questions

Any questions such as "Are you married?" or "Do you have kids?"





# More resources on developer interviews

The guerilla guide to interviewing by Joel Spolsky

 $4\,Ways\,to\,upgrade\,your\,interview\,process\,for\,developers$ 

Is whiteboard coding inappropriate during interviews?











# Are you hiring developers?

Get your opportunities in front of the largest audience of software developers on Stack Overflow, the platform they use and trust every day.

# Talk to an expert →





