

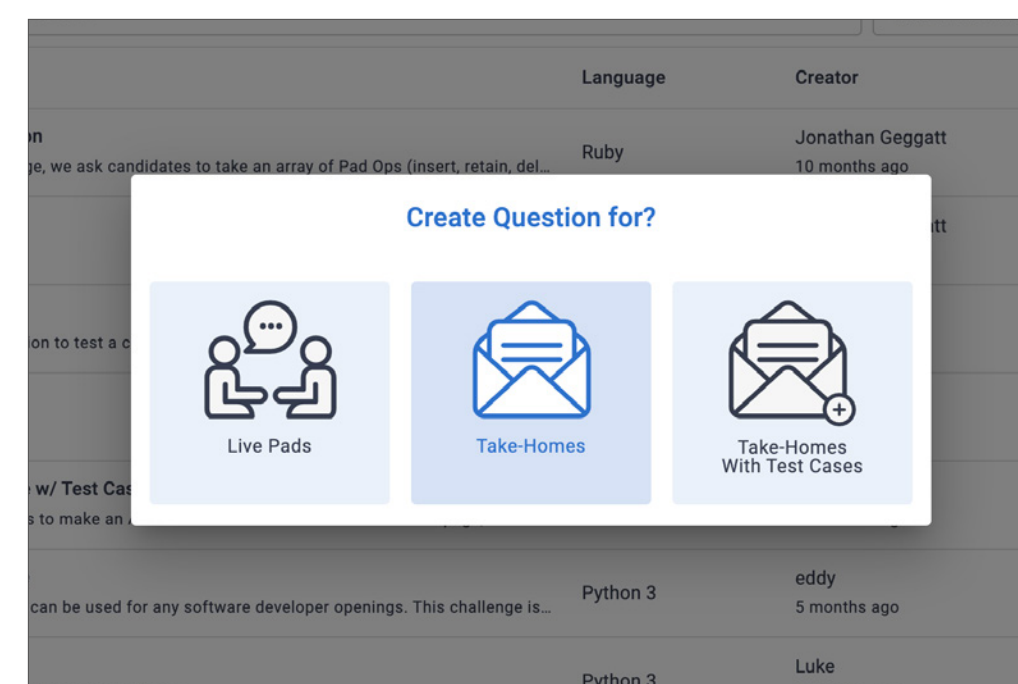
Create an Inclusive Interview Experience with CoderPad

CoderPad's platform enables consistent, fair technical interviews by giving developers the environment they need to do their best work. It offers critical tools to evaluate candidates effectively and help reduce bias.

Skills-Based Screening

At the start of many interview processes, there's a dependence on resumes to determine whether or not a candidate is right for your team. The problem is, a list of a candidate's previous jobs and educational background does not tell you whether or not they possess the technical skills needed to be successful in the role. Instead, it introduces assumptions and possible bias into the process.

Evaluate all applicants equally by looking at their work samples. Create a project that can be made available publicly for all interested applicants with our [Public Take-Home](#). Add the url to your Career page, University Recruiting campaign or send it to a batch of candidates. This enables you to replace a resume review with a skills-based assessment, and also provides an alternative path for candidates who may not want to fill out a job application but do want to show off their skills.

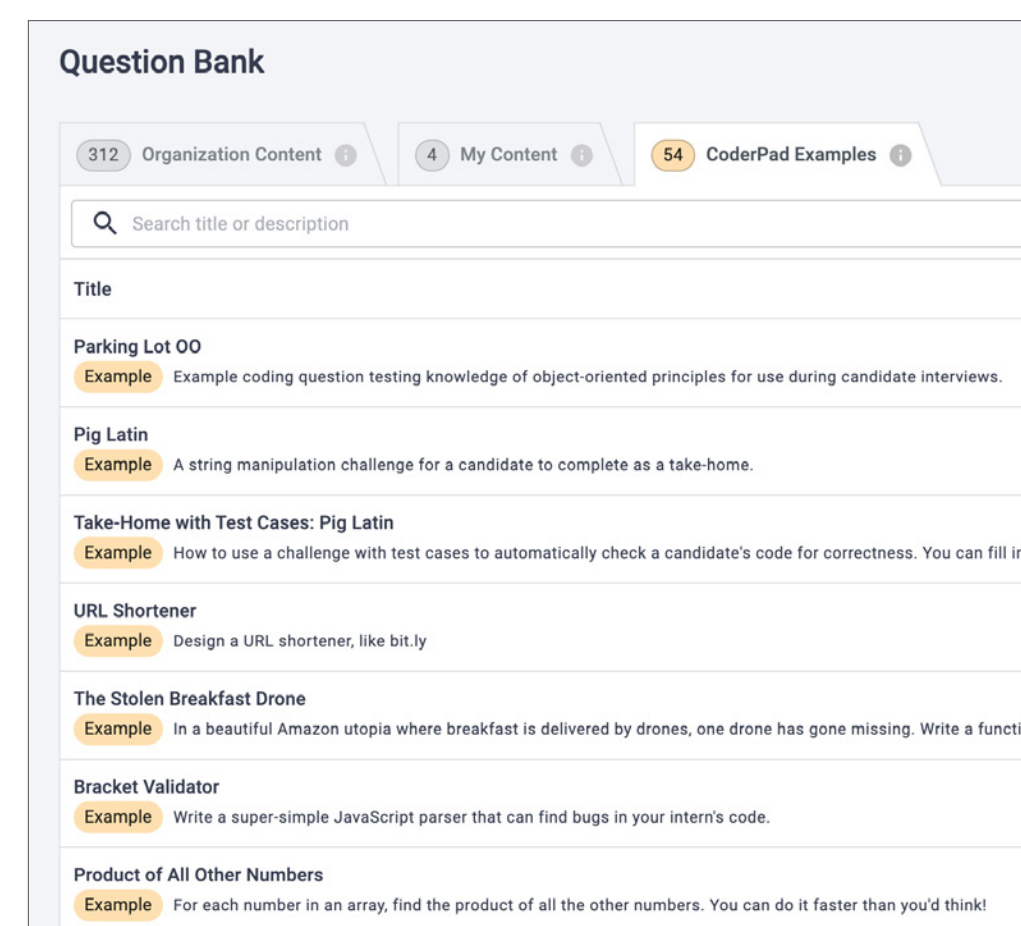


Standardized Candidate Experience

To ensure a fair playing field for every candidate, technical interviews should follow a standardized experience. Creating a structured interview process with evaluation criteria will help you objectively assess each candidate and reduce bias.

Ask the Right (Standard and Specific) Questions: Centralize your interview questions for every level and type of job within your [Question Bank](#). Create these questions based on realistic projects a person would do in this role at your company to see if their skills fit your needs - and review the questions on a regular basis to make sure they're still applicable. After the interview, make sure the hiring manager and interviewers are on the same page by reviewing the interview [Playback](#).

Set Clear Evaluation Criteria: Have your team follow a standard process for every interview so that they all know what "good" looks like. Implement a rubric-based grading system and give interviewers clear solutions to each question. Implementing these guidelines while using [CoderPad](#) throughout your hiring stages ensures you don't over-index on candidates who are good at navigating interviews but don't have the technical chops.



The Right Environment

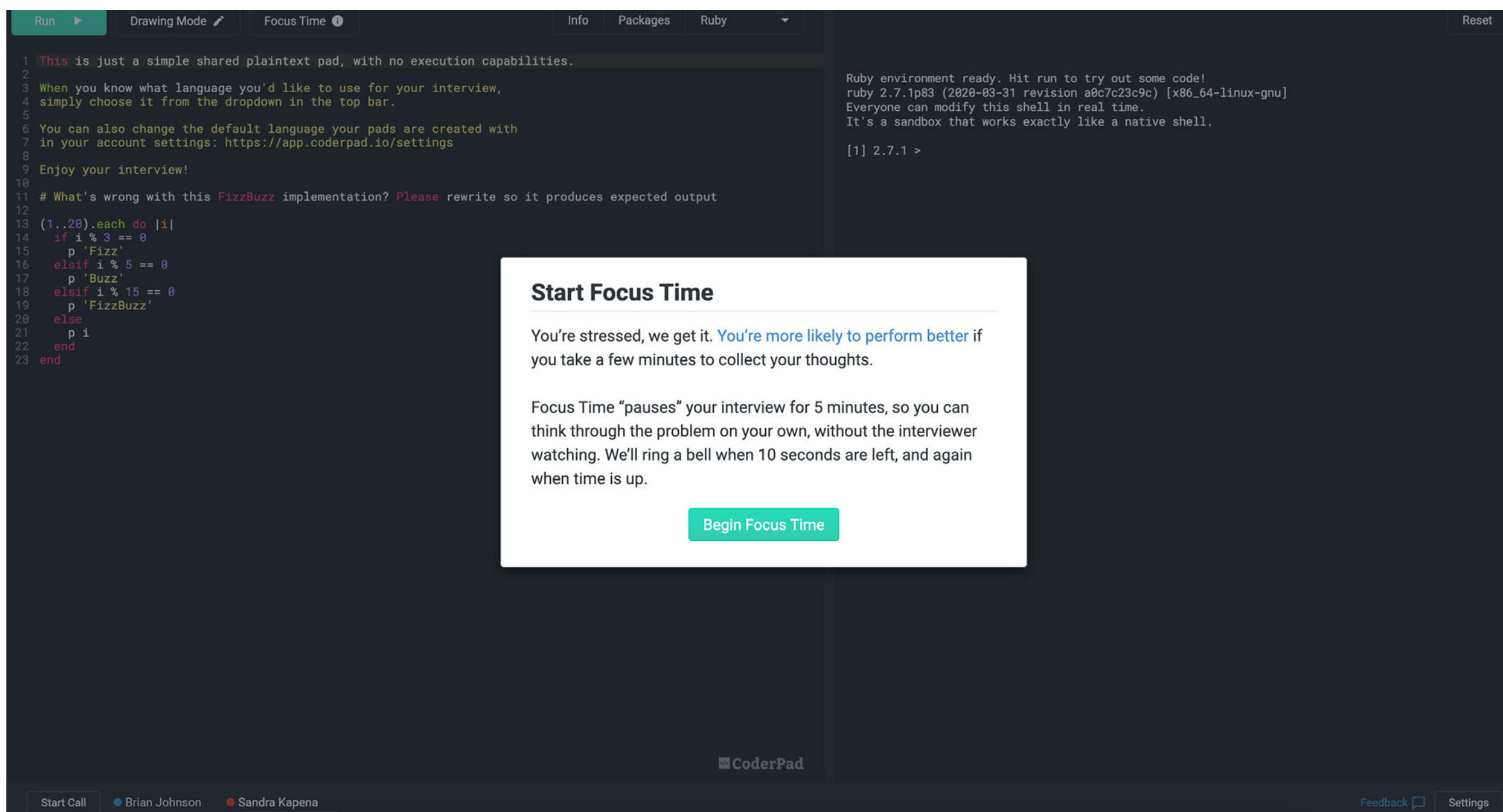
Developers need to feel comfortable in their environment to do their best work. This means the coding platform for each interview must be familiar and intuitive, instead of something with a steep learning curve. It also means giving them the space and comfort to show off their skills - not participate in a stress test.

Ensure Realistic Coding

Give candidates the ability to show off their skills - and give you a signal that they're the right fit - by giving them a familiar, [easy-to-use environment for coding](#). CoderPad's familiar code editor allows devs to feel at home, with features they expect to have in an IDE. Candidates and interviewers alike can customize their editor settings in CoderPad - giving them a platform where they can do their best work.

Provide a Comfortable Interview Experience

CoderPad has worked with leading academics to create [Focus Time](#), a feature that gives candidates 5 minutes of quiet time to gather their thoughts before they begin working on a solution. It gives them the opportunity to settle nerves, get organized, and think about the technical problem. [Studies have shown](#) that candidates in a traditional interview who are watched by an interviewer performed half as well as those who solved the problem alone with space to think.



The screenshot displays the CoderPad web interface. On the left, a code editor shows Ruby code for a FizzBuzz implementation. The code includes comments and instructions for the user to rewrite it. On the right, a terminal window shows the Ruby environment ready for execution. A modal dialog titled "Start Focus Time" is centered on the screen, explaining the feature and providing a "Begin Focus Time" button. The interface also shows a top navigation bar with options like "Run", "Drawing Mode", "Focus Time", "Info", "Packages", and "Ruby". At the bottom, there are buttons for "Start Call", "Feedback", and "Settings", along with user avatars for Brian Johnson and Sandra Kapena.

```
1 This is just a simple shared plaintext pad, with no execution capabilities.
2
3 When you know what language you'd like to use for your interview,
4 simply choose it from the dropdown in the top bar.
5
6 You can also change the default language your pads are created with
7 in your account settings: https://app.coderpad.io/settings
8
9 Enjoy your interview!
10
11 # What's wrong with this FizzBuzz implementation? Please rewrite so it produces expected output
12
13 (1..20).each do |i|
14   if i % 3 == 0
15     p 'Fizz'
16   elsif i % 5 == 0
17     p 'Buzz'
18   elsif i % 15 == 0
19     p 'FizzBuzz'
20   else
21     p i
22   end
23 end
```

Ruby environment ready. Hit run to try out some code!
ruby 2.7.1p83 (2020-03-31 revision a8c7c23c9c) [x86_64-linux-gnu]
Everyone can modify this shell in real time.
It's a sandbox that works exactly like a native shell.
[1] 2.7.1 >

Start Focus Time

You're stressed, we get it. [You're more likely to perform better](#) if you take a few minutes to collect your thoughts.

Focus Time "pauses" your interview for 5 minutes, so you can think through the problem on your own, without the interviewer watching. We'll ring a bell when 10 seconds are left, and again when time is up.

[Begin Focus Time](#)