

Lesson

Monday

Intermediate JavaScript (/intermediate-javascript)

/ Test-Driven Development and Environments with JavaScript

(/intermediate-javascript/test-driven-development-and-environments-with-javascript)

/ Asking and Listening

Text

This lesson is part of our regular Diversity, Equity, and Inclusion curriculum. This week we will discuss two of the most powerful tools at our disposal: **asking** and **listening**. These are essential soft skills that will help us be more inclusive, and are also instrumental to getting jobs.

Asking Questions

When we are learning how to code and spending many hours every day problem-solving, it can be easy to get overly focused on trying to have all the answers.

This can be particularly problematic when it comes to inclusion. In order to work well with others, the questions are often more important than the answers.

Here's an example. Let's say that two people are working on a coding problem. One person, Tim, is driving (using the keyboard while the other person observes) while the second person, John thinks he has an answer. Which of the following might be a good approach?

A. John says "Let me try something" and starts typing without waiting for an answer.

B. John says "You should try this" and then tells Tim what to type.

C. John asks "I have an idea. Can I run it by you?"

It should be clear why option A and B are problematic. Tim is driving — and he may want an opportunity to try out a solution or solve the problem on his own — but John simply takes over without checking in first. There's a real risk that Tim will feel shut down. If Tim is a member of an underrepresented group, such as Latinx or trans people, that John is not a member of, then Tim may be left wondering if John disregarded Tim's knowledge because of his race or gender. John may come across as patronizing or bullying as a result while Tim might feel hurt or disengaged.

Option C is different because John asks his pair a question. Ultimately, it's more important to be collaborative in this situation than it is to immediately have the answer. In a real-world workplace, having good working relationships and being collaborative isn't just inclusive — it will increase productivity and keep you and your coworkers happy.

On the other hand, John having all the answers right away may save a little bit of time — at least at first — but in the long term, if he continues engaging in this behavior other employees will become less collaborative and less likely to share their own solutions, and avoid working with him.

Option C may seem like the obvious best solution to the scenario above. However, we do see students engaging in option A and B in the classroom. It's all too common to see students sharing answers even where they aren't wanted, which comes across as patronizing behavior. In the case of a man explaining something to a woman or non-binary person, it comes across as "man-splaining."

A better approach is to start with questions, not answers. Here are some examples that are useful at Epicodus and beyond:

- "What do you think?"
- "I have an idea. Can I run it by you?"
- "Do you mind if I try something?"

Taking this approach will improve our pairing skills, help us prevent behavior that shuts our peers down, help us avoid microaggressions, and make us better future employees.

Of course, asking is only part of the equation. It's even more important to listen.

Listening

Let's take another look at the scenario from before. John says "I have an idea. Can I run it by you?" In response, Tim says, "Not quite yet. I want to try something first." John insists on trying his idea anyway and takes the keyboard from Tim.

In this scenario, John asks a question but he doesn't listen to or heed Tim's answer. The end result is the same — perhaps even worse — than if he'd just taken the keyboard without asking. Tim feels shut down. He may not be invested in working with John anymore. He may even decide not to pair with John again in the future.

The issue here is that John didn't listen to Tim's response. Listening is even more important than asking.

In order to be better listeners, we can do the following:

1. **Make sure we understand what's being said.** Here are some questions we can ask to ensure we don't just listen but also understand what others are telling us:
 - "You just said _____. Is that right?"
 - "Did I understand that correctly?"
 - "Would you mind clarifying that for me?"
2. **Respond or take action based on what we've been told.** For instance, if we ask a question, we need to be prepared to listen and take action based on the response. In the example above, when Tim says he wants to try another approach, John's actions demonstrate a disregard for his pair's response. The better course of action here would be for John to wait until it's his turn to drive or until Tim lets John know that he'd like to hear his answer.

Asking and Listening Can Help Us Avoid Common Pitfalls

Here are some issues that can come up if we don't ask and listen.

- **Cutting someone off or interrupting.** The underlying message here is that what the other person has to say isn't important. This also applies to driving when coding as well — we can interrupt by typing when it's our pair's turn to drive.
- **Giving an opinion without being asked.** This often comes across as being patronizing or "splaining."
- **Being offensive or hurtful.** If we aren't listening, it may well seem that we don't have regard for the other person. The underlying message here is that the other person's feelings and thoughts don't matter.
- **Appearing arrogant and out of touch.** No one has all the answers. Acting as if we do — or disregarding the ideas of other people — suggests that we think more highly of

ourselves than others.

- **Making assumptions or miscommunicating with others.** Communication breakdowns can lead to buggy code, hurt feelings, and other issues — and it's unlikely that others will want to pair with us or work with us again if they have a bad experience with us.

It should be clear that poor listening skills can lead to a wide variety of problems, microaggressions, and general miscommunication issues.

We can solve most problems by asking questions and being good listeners.

Ready to Write Your Reflection?

There is a reflective assignment for this lesson. If you are ready to write your reflection, head on over to Epicenter to find the prompt. If you are logged in to Epicenter, you can access the prompt by navigating to this link:

🔗 **Reflection Prompt: Asking and Listening**
(<https://epicenter.epicodus.com/journals?title=Asking+and+Listening>)

Otherwise, you can find detailed instructions on accessing the reflection prompts in the DEI Reflective Assignments (<https://www.learnhowtoprogram.com/introduction-to-programming/getting-started-at-epicodus/dei-reflective-assignments#finding-the-reflection-prompts>) lesson.

Do you have feedback?

We want to hear about your experience of the DEI curriculum. We outline all of the ways you can give feedback in the student handbook (<https://www.learnhowtoprogram.com/introduction-to-programming/getting-started-at-epicodus/student-handbook#giving-feedback>).

[Previous \(/intermediate-javascript/test-driven-development-and-environments-with-javascript/improving-test-reports-adding-test-coverage-information\)](#)

[Next \(/intermediate-javascript/test-driven-development-and-environments-with-javascript/test-it-out-sudoku-checker-project-euler\)](#)

Lesson 33 of 49

Last updated more than 3 months ago.

[disable dark mode](#)



© 2023 Epicodus (<http://www.epicodus.com/>), Inc.