



( / )

Curriculum

**SE Foundations** ^

Average: 108.76% v

# Pair Programming - How To



Regardless of if you think you understand something, pair programming is an invaluable industry skill that you need to practice

Perhaps, it seems obvious what you need to do, but ensure you know what you're doing by learning through the examples below.

[Help](#)

Let's say we have two students, Kristen and Stuart. Stuart is trying to understand a few things that Kristen wants to help with. Here's how it'd break down:

Kristen is not allowed to touch the computer; instead, they need to make sure that Stuart goes through The ALX Framework.

Kristen should try to prevent Stuart from searching Google if the answer can be found in the man or documentations provided on the Intranet.

If Stuart struggles too much solving one task, Kristen is allowed to use the computer and code the solution in front of Stuart, and explain the thought process.

Kristen should make sure that Stuart understands everything, though. When done, Kristen deletes all the files containing the solution, and Stuart tries again.

### If you're trying to help someone:

- When explaining a concept, make sure you give examples to illustrate. Try to use whiteboards as much as you can.
- Be patient and understand that people think differently about the same problem. Sometimes, something will seem obvious to you, but to someone else, it may not be.
- **Most of the time, if you explain the same concept twice and your peer doesn't get it, it's because you didn't explain it well enough :) It's your fault, not theirs.**
- Try to rephrase, break it into smaller pieces, find new examples
- To ensure someone understands a concept, it is not enough to ask "do you understand". You should start with this question, yes, but you should ask them to solve a similar small task using the same concept. A good way to test is to ask a knowledgeable peer to explain it again to you.

For Instance, if [following our prior example] Kristen explains permissions / modes, the questions should be:

- Do you understand?
- Yes
- Great! Explain it back to me now please :)
- ...
- Great, explain what does `rw-r--r--` mean
- ...
- Great, so what would be the octal notation of `rw-r--r--`?
- ...
- Great! what would be the permissions of a directory with a mode of 755?
- ...



## Pair programming Continued ...

When a peer receiving support completes all tasks demonstrated by the peer helping, they should work together on the remaining tasks (including the advanced tasks). Same rules apply: \* Read the documentation

- Don't code before you have a potential solution
- Explain out loud your thought process
- Etc..

For each task, switch the students who is typing. The student needing support should start. When switching, the new student needs to code the same task again. **No copy-paste.**

Example: There are 3 tasks remaining: T0, T1, T2, and Kristen and Stuart are still our examples of collaborating students.

- Stuart and Kristen solve all tasks together, but:
- Stuart starts typing for T0 and when done pushes on his github.
- Then Kristen types again the solution of T0 and pushes to her github.
- Then Kristen types for T1, pushes on her github.
- Then Stuart types again the solution of T1, pushes, and starts T2.
- When Stuart and Kristen found the solution for T2, Stuart pushes to his github.
- Then Kristen types the solution again and pushes on her github.

Copyright © 2024 ALX, All rights reserved.

