Peer to Peer

Pair Programming with Epic React

Two types of Learning

Easy - passive

Watching a tutorial or reading an article.

Hard - active

Coding challenges

Projects with code reviews

When learning is hard

We are tempted to avoid the struggle by:

1. By giving up
2. By focusing on researching and understanding every last detail of the topic but not actually coding
3. By making it work using code snippets from Stack overflow with no idea *how* the snippets work

The good and the bad

We don’t want to give up completely but taking a break and coming back is good.

Researching it and making it work are vital parts of being a developer.

But we don’t want these strategies lead to avoidance.

We don’t want to get stuck doing only one of them and don’t balance both.

Balance with Pair Programming

Ideally we want to balance ‘research’ with ‘making it work’.

How can we do so?

With pair programming.

Essence of Pair Programming

Navigator (the explainer)

Driver (the implementer)

It requires you to explain your strategy to your partner with clarity and precision - improving your technical communication and underlying problem-solving approach.

Try and switch driver/navigator roles every 25 minutes

More specifically, how Do You Pair Program?

One person drives, controlling the keyboard and mouse, and the other person watches and talks.

How can this be better than coding solo?

Two heads are better than one when coming up with solutions.

Programming is less about writing massive amounts of code and more about solving difficult problems.

Pairing lets you learn from each other.

Usually, each person in a pair will have different knowledge and different strengths.

Continued...

When pairing, you can hold each other accountable to write high-quality code.

When you work alone, you're often tempted to take shortcuts that will come back to haunt you. For example, you might not adequately test your code to make sure it works.

You're much more productive.

You don't check your email, social networks, news, text messages, and so on when you're pairing.

You can spot more mistakes easily

When you watch somebody else code, it's easier for you to see their mistakes and catch their bugs

PP has advantages but also can be frustrating

You'll often be working with someone who is either more or less experienced than you.

You'll often feel like you're slowing down your pair or they're slowing you down.

You might want to explore how something works while your pair wants to focus on actually finishing the project at hand.

You might run into a difficult bug and you'll have different ideas about the right way to try to fix it.

Communication is key

Try and communicate before diving into coding.

Take a deep breath, take a break if you need to, and then talk it through with them. It's tough, but it's better than suffering through the class.

Remember that everybody will have difficult pairs, especially at first.

It’s an additional skill that gets better with practice

Celebrate your wins

Refrain from value judgements

Please refrain from making value judgments about your peers (or yourself!) based on your perceived difference in skill sets.

It’s important to work with a wide variety of people, including people at your same level of understanding.

Make the best of Pair Programming

Take the opportunity to really solidify the concepts and technical terminology at hand.

You can offer to share your strategies for practice and understanding of concepts.

Teach each other the concepts that each of you may be struggling with.

Questions are always good!

Share your ideas freely

Take the time to develop and discuss your ideas and problem solving steps for the task at hand.

Continued...

It's extremely common for students to not understand something the first time they read about it.

Likewise, it's common to need time, practice, and repetition to really understand concepts.

Students often revisit concepts later and work on certain concepts over a few course sections.

So, if you don't understand something yet, you will soon!

Good Luck! Happy Coding!

Stay positive, have fun, and take breaks!

Remember, being a developer is not about learning a fixed set of skills that you can apply for the rest of your career.

The languages, tools, and approaches you'll learn here are much less important than the general skill of solving problems.

Epic React Testing Workshop

Break up in pairs

Start at the beginning or under the Mocking with Jest portion