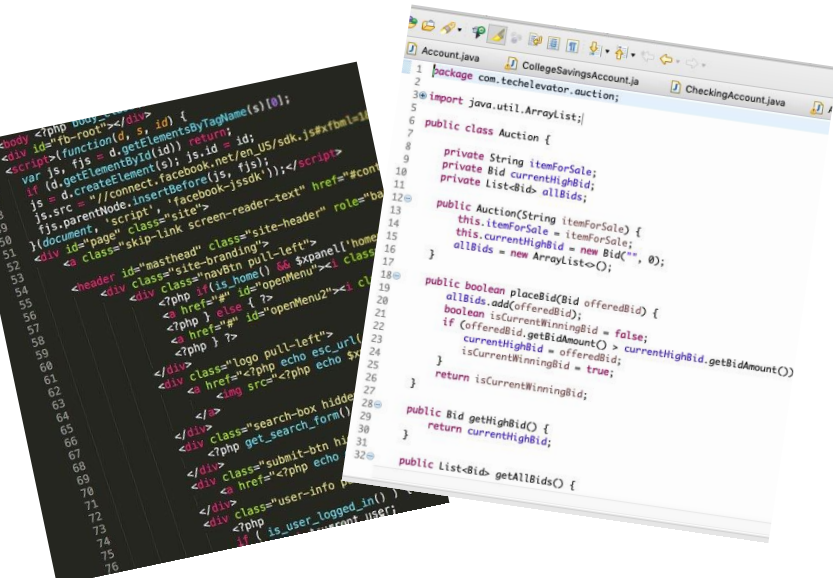


A person in a dark suit and white shirt is giving a thumbs up gesture. The background is dark and out of focus. A semi-transparent grey rectangle is overlaid on the image, containing the title text in white.

# **How to Become an Amazing Pair Partner**

# Before you jump into code....

- Do the small talk
- Agree on the physical environment
  - Where you will work
  - Light vs Dark
  - Keyboard shortcuts and mouse behavior



## Identify team & individual goals

Applies to mob programming too!!

## Establish a rhythm

Drivers drive. All Navigators review each line of code as it is typed and consider “strategic” direction of the work.

(Allow the driver to finish typing the current block of code before pointing out a typo)

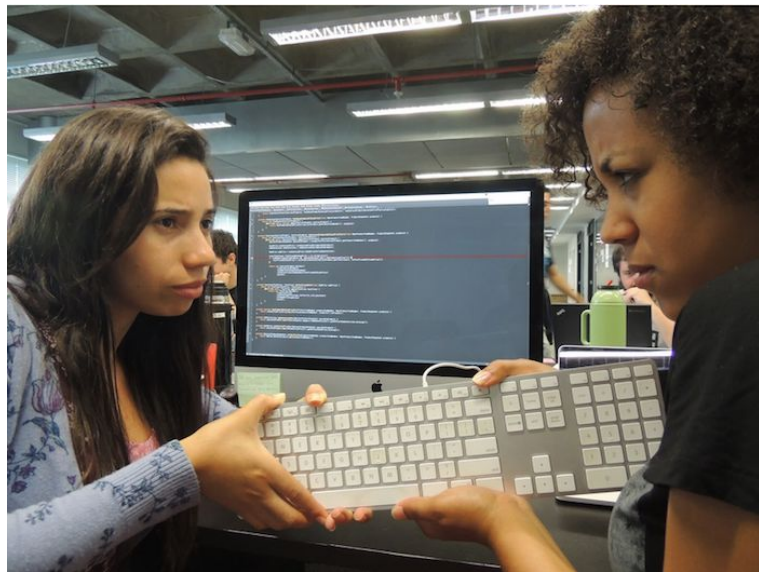
Change driver/navigator ~ 30 min and commit frequently.

TTD - one writes the test, the other solves

## Think out loud

## Slow down

## Make an effort to compliment as much as possible



When talking about code, always refer to line number and file name

Take a look at the following question:

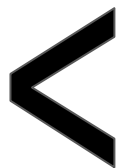
What's the value of the variable?

Compare the content of that question to the following question:

“What is the value of the listCount at Line 232 of the file, MyFile.java?”

When proposing an idea, articulate the benefit of the idea.

Opinion



Idea  
Benefit



- Put competition aside
- it is very important for each pair to hear the other completely and express their ideas in a calm and respectful way
- If the pair can't agree on the path to a solution, calling a third person to vote on one of the given solutions may help

When feeling ill at ease, say so

When on the defensive or frustrated we are focused on that uncomfortable feeling, and not on better ideas.

Remember we are all human -

You can say something as simple as, “Dude, I know you probably didn’t mean it, but calling my idea lame really hurt. Maybe moving forward you can just say that my idea has some shortcomings we need to address.”



# What should you do if your partner has expressed their frustration?



Take a 15 min break by yourselves to hear what your partner said

Try not to use this time to 'vent' to someone else or build your case

See the human you are working with

Review the goals of project/team to yourself

Practice saying, "I am sorry."

Avoid working alone/reworking code

Embrace challenges

Celebrate each small success

High-five / synchronized backflip or chair-spin  
coffee break / snacks

Take a minute at the end of a coding session to review your commits and discuss what's next.





# After Tech Elevator....

As a Junior Developer you will:

Be inexperienced

Lack confidence

Hesitate to make a suggestion

Have minimal knowledge of the codebase

Focus of learning:

Codebase, terminology, architecture, IDE shortcuts

Senior Developers you work with will::

Be an expert

Be confident

Happily suggest solutions

Knows the codebase very well

Focus of learning:

Best practices, trends, system performances

## Pair-programming from a beginner's perspective

### Pain points of the Junior

I am slowing him down

Sometimes I don't understand what he is saying

I ask questions all the time

If don't get to drive, I struggle to stay engaged

If I disengage from the session regularly, I am not learning

If I am not learning, I will remain a slow pair for a long time



## Pair-programming from a beginner's perspective

### Pain points of the Senior

He is slowing me down

We've been discussing this for 10 minutes - it will take me just 2 seconds to type the thing in and move on

This is a one-way conversation - I seem to talk and explain all the time

What kind of pairing is this?



# The more advanced partner should.....

Don't over-optimize

Ask questions

“How about we extract a method?”

Admit when you don't know. Show them how to look it up.

Admit when you're stuck. Show them how you get unblocked.

Admit when you're tired. Show them the worthiness of breaks.

## All pair partners need...

- Opportunity to try things out
- Permission to make mistakes
- Constructive feedback
- Friendly disposition from their pair

For more information

<https://blog.rapid7.com/2017/01/27/5-rules-of-pair-programming-etiquette/>

<https://gds.blog.gov.uk/2018/02/06/how-to-pair-program-effectively-in-6-steps/>

<https://speakerdeck.com/itsyganok/pair-programming-from-a-beginners-perspective>