Interviewee 2: [redacted]
Undergraduate: [redacted]
Currently taking Masters.
Works: CN (Stack Angular + Spring)I - Business Consultant - Communication between developers and customers. Programs on his own projects (in JS)
JS experience: 4 years
Other languages: Java, C#, Python.
Writes new modules more than maintains legacy modules.
JS Pros: Very practical language. Abundance of frameworks, especially TypeScript, which allows us to program in JS as though it was typed.
JS Cons: Too much freedom/flexibility. It can open room for unaware/inexperienced programmers to make lots of mistakes and/or ugly/unreadable code. - *"Programadores que não estão muito atentos a design patterns e, falando a grosso modo, a programar o negócio direito, acaba possibilitando muita coisa feia, coisa difícil de ler depois, de entender o que está sendo feito."*
Does JS lead to hard-to-understand code? In the absence of design patterns in the project, yes.
Any particular JS construct that is hard to understand? To whoever does not deeply understand it, some functions are not so intuitive (functional constructs, such as map.). In contrast to Java and Python, yes. (Adriano's opinion here: functions like bind, call and apply can be very difficult to understand in JS. We can study it later)

Atom	Preferred version
Arithmetic as Logic	Indifferent
Assignment as Value	Without atom
Automatic semicolon insertion	Indifferent
Comma Operator	Without atom
Ternary operator*	Without atom
Implicit predicate	Without atom
Logic as Control Flow	Indifferent
Omitted curly braces and Indentation	Without atom
Post increment	Without atom
Pre Increment	Without atom

*Again here the interviewee prefers to write the ternary operator, but finds side B more readable. In the TCC research, code without the ternary operator had a positive delta of 9% in correctness, but participants generally took the same amount of time to evaluate the code.

Hypothesis: experienced devs prefer to write it this way, regardless of it being overall more complicated to newcomers In other words, developers do not care enough about the future people who might come into contact with the code.
Also, the participant took a LONG time (40 seconds) to evaluate this question.
“Quando eu programo eu faço o ternário, [...], só que, para ser verdadeiro, eu continuo achando que o B é mais fácil de entender para quem não conhece tanto.”

This interviewee tends to prefer to write less verbose constructs, even if they are harder to understand for people who read the code in the future.

Adriano’s suggestion: In the future, we can study a corollary of the ternary operator, which to me is crazy: **ternary operator chaining**. It is not uncommon in JS codebases, and I personally met a dev who basically wrote all his branching logic using them. To me they make the code almost impossible to understand.