Interviewee name: [redacted]
Study: MsC
Works: [redacted]
JS experience: worked 13 years ago. Some contact with it currently
Other languages: Java, PHP
When developing in JS, developed new modules, as well as maintained old ones
JS Pros: Speed. No longer necessary to perform AJAX requests manually
JS Cons: None. JS is excellent. Although it is verbose (pure JS), frameworks have made it easier over time. JS code is a bit harder to debug than other compiled languages.

Note: There is a long/ongoing discussion of whether JS is compiled or interpreted. Kyle Simpson (https://github.com/getify/You-Dont-Know-JS/blob/2nd-ed/README.md), who wrote what is, to me, the best book series in JS available, approaches it as a compiled language. Note that JS performs 2 parse phases, which actually hinder its performance.

Do you think JS leads to hard-to-understand code? No.
Any particular constructs that can lead to difficulties in understanding the code? Callbacks, especially for beginners.

Note: this problem with passing a function (callback, in this case) is not exclusive to JS. It can happen with any language that supports functional programming, which is the case for JS (it was strongly influenced by functional programming since its inception - you can write *pure* functional code in JS).

Atom	Preferred version
Arithmetic as Logic	Without atom
Assignment as Value	Indifferent
Automatic Semicolon Insertion	Without atom
Comma Operator	Without atom
Ternary Operator	Without atom
Implicit Predicate	Indifferent
Logic as Control Flow	Indifferent
Omitted Curly Braces and Indentation	Without atom
Post Increment	Without atom
Pre Increment*	Indifferent

*Interviewee gives the odd opinion that pre increment is simpler, and less error prone. This is controversial, as in pre-increment, increment happens before evaluation, which can lead to a lot more mistakes.