

Interviewee 1: [REDACTED]

Undergraduate: Computer Science [REDACTED]

Specialization Data Science [REDACTED]

Works: Stefanini - Developer

JS Experience: 6 years

Other languages: C, C++, Java, Assembly, Kotlin

Writes new modules, and maintains current modules (**Note: features != modules**)

**JS Pros:** Syntax similar to C, C++, C#. Quick to learn, due to several ways to write the same thing.

[REDACTED]

Differential: Runs exclusively on client side (**not true**)

ES6 abstracted away lots of complex stuff

**Cons:** Too many updates. Today is no longer JS, it is ECMAScript (**It has been like this for a long while**). Interviewee suggest discontinuing old JS, but that makes no sense as far as backwards compatibility is concerned.

Too many data types that can come from uninitialized objects. **Difference between null, undefined, NaN.**

Do you think JS leads to hard-to-understand code? It depends on the developer.

**Prototype-based language can be difficult to understand.** (This is a very good point. It is indeed difficult to understand JavaScript's prototype chain.)

ES6 added an abstraction layer on top of the prototype chain (ES6 classes).

Any specific construct that makes JS code tough to understand? **Prototype.** [REDACTED]

[REDACTED]

Construct	Preferred version
Arithmetic as Logic	Without atom
Assignment as Value	With atom
Automatic Semicolon Insertion	Without atom
Comma Operator	Without atom
Ternary Operator	With atom*
Implicit Predicate	Without atom
Logic as Control Flow	Without atom
Omitted Curly Braces & Indentation	Without atom**

Post Increment	Without atom
Pre Increment	With atom

