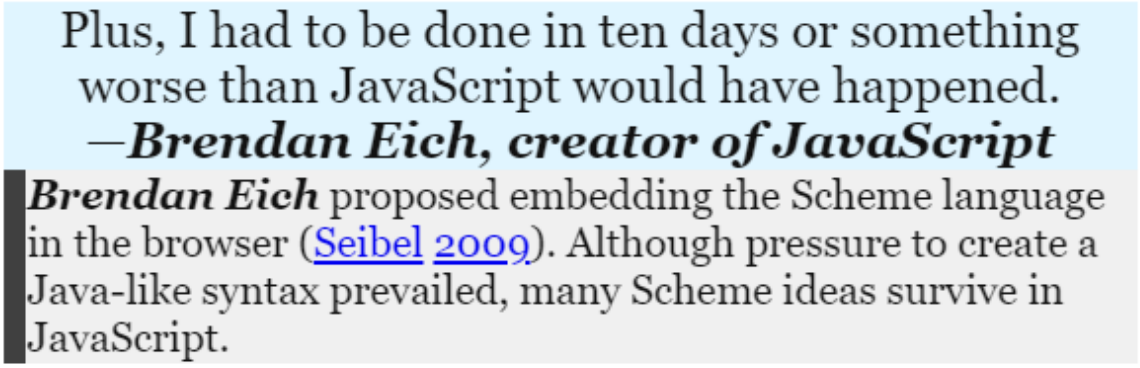
Interviewee: [redacted]
Licentiate in Computing
Works at [redacted]
JS experience: more than 15 years since first contact. 3 years professionally
Other languages: Java, PHP.
Builds new and maintains existing modules.
JS Pros: The ecosystem. Very practical and flexible language. Easy to set up an environment. Weak typing. Allow developers to start from basic and evolve to complex things.
JS Cons: When it is in the browser, there can be compatibility issues. Technically, it was not very well thought. Object comparison. Language is too permissive (due to weak typing)
Adriano's comment: Screenshot below adds a quote from Brendan Eich, the creator of JavaScript (Extracted from book: Engineering Software as a Service, by Armando Fox and David Patterson):



Do you think JS leads to hard-to-understand code? It can, but I have never seen any language that inhibits this. Most of the JS code I use I understand more easily than in other languages. "Na minha opinião o código depende mais do programador do que da linguagem em si. [...] eu já vi códigos complexos em todas as linguagens com que trabalhei"

Any JS construct that can be particularly difficult to understand? Excessive inheritance between 'classes'. Excessive usage of functional constructs (map, reduce). Programmer's misusing JS's weak typing (but this is not technically a construct)..

Atom	**Preferred version**
Arithmetic as Logic	Without atom
Assignment as Value	Without atom
Automatic Semicolon Insertion	Without atom
Comma Operator	Without atom
Ternary operator	Indifferent
Implicit predicate	Without atom

Logic as Control Flow	Without atom
Omitted Curly Braces and Indentation	Without atom
Post Increment	Without atom
Pre Increment	Without atom

Any relevant remarks about JS? NaN comparisons.

Adriano's notes:

NaN === NaN -> false
Number.NaN === Nan -> false
isNaN(NaN) -> true
isNan(Number.NaN) -> true

```
Function valueIsNaN(v) {
        return v !== v;
}
```

valueIsNaN(1) -> true
valueIsNaN(NaN) -> true
valueIsNaN(Number.NaN) -> true

Yes, this is messy.