

Enhancing Your Libraries' Security and Performance through Macros

Veit Heller
Clojure Meetup Berlin
October 30, 2019

“Just remember that macro programming is not about style; it is about power.”

— Doug Hoyte, Let Over Lambda (Chapter 3)

```
(defmacro fmt [s & args]
  (let [a (dec (count (filter #(not= "" %)
                              (str/split s #"%"))))
        b (count args)]
    (if (= a b)
        `(apply format ~s '~args)
        (exc "arguments to format don't match (expected " a
              ", got " b ")")))))
```

Listing 1: A safer format

```
(defmacro defmacro! [name args & body]
  (let [rm (into {}
                  (map (fn [s] [s `(quote ~(gensym))])
                       args))])
    `(defmacro ~name ~args
      `(let ~~(into [] (mapcat reverse rm))
        ~(clojure.walk/postwalk-replace ~rm ~@body))))))
```

Listing 2: A macro utility—or, as Alan Perlin said: “Syntactic sugar causes cancer of the semicolon.”

- ▶ Greg Hendershott: Fear of Macros
- ▶ Doug Hoyte: Let Over Lambda
- ▶ Colin Jones: Mastering Clojure Macros
- ▶ Paul Graham: On Lisp
- ▶ These slides: <https://github.com/hellerve/talks>
- ▶ A series of blog posts on Scheme macros:
<https://blog.veitheller.de/scheme-macros>

Thank you!

Questions?

Slides at <https://github.com/hellerve/talks>