

Clojure programming language & it's ecosystem

Marius Rabenarivo

The logo for the Developers Conference 2024 is rendered in a pixelated, retro style. The text "DEVELOPERS" and "CONFERENCE" are stacked on two lines, with "2024" centered below them. The letters are primarily yellow and orange with a red outline, and the entire logo has a blue drop shadow.

**DEVELOPERS
CONFERENCE
2024**

What is Clojure?

- LISP



Figure 1: Lisp Cycles, <https://xkcd.com/297/>

- Rich Hickey



Figure 2: Rich Hickey at the first Clojure Conj in 2010, <https://creativecommons.org/licenses/by-sa/2.0/>

What is LISP?

Y defun \mathcal{Y} (f) ((λ (g) (funcall g g)) (λ (x) (funcall f
(λ (rest a) (apply (funcall x x) a))))))

F defun fib (n) (funcall (\mathcal{Y} (λ (f) (λ (n a b) (if (< n I)
a (funcall f (I- n) b (+ a b)))))) n o I)

F defun fac (n) (funcall (\mathcal{Y} (λ (f) (λ (n) (if (zerop n) I
(\times n (funcall f (I- n))))))) n)

$\hat{=}$ (mapcar #'fib '(i ij iij iv v vi vij viij iε))
(i i ij iij v viij ϵ ij $\epsilon\epsilon$ i $\epsilon\epsilon\epsilon$ iv)

$\hat{=}$ (mapcar #'fac '(i ij iij iv v vi vij viij iε))
(i ij vi $\epsilon\epsilon$ iv $\epsilon\epsilon\epsilon$ $\overline{D}\epsilon\epsilon\epsilon$ $\overline{v}\epsilon$ l $\overline{\epsilon D}\epsilon\epsilon\epsilon\epsilon$ $\overline{\epsilon\epsilon\epsilon v i m m D}\epsilon\epsilon\epsilon\epsilon$)

Figure 3: Y Combinator Codex by emacsomancer.

What is LISP?

- LISP Processing language
- 2nd oldest High-Level Language after FORTRAN
- 1958 - John McCarthy

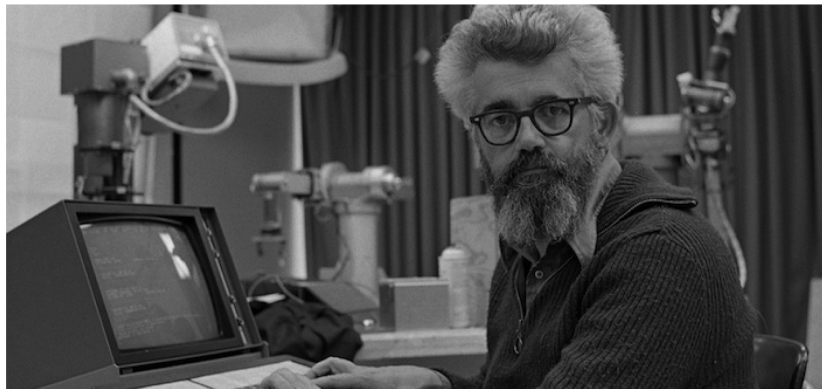


Figure 4: John McCarthy at work in his artificial intelligence laboratory at Stanford

Why Clojure?

- Designed with simplicity
- Functional Programming
- Data Oriented



Figure 5: Code? Data? Code? Data?

Why Clojure?

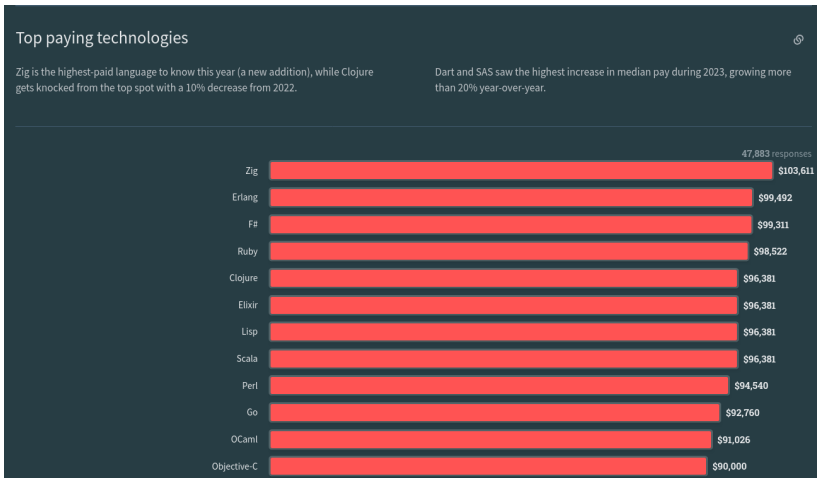


Figure 6: 2023 Developer Survey, Top paying technologies

Source: <https://survey.stackoverflow.co/2023/#section-top>

Why Clojure?

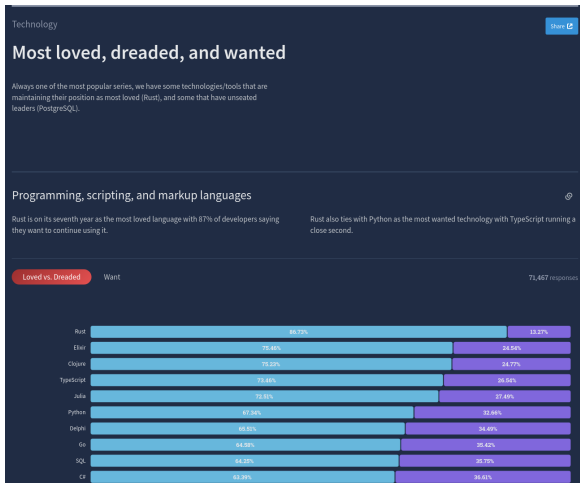


Figure 7: Stack Overflow Developer Survey 2022, Most loved, dreaded, and wanted

Functional Programming

- Immutability
- Referential transparency
- Easy to think about systems
- Parallel and concurrent programming

Where to go from here?



Figure 8: Clojure for the Brave and True, Daniel Higginbotham