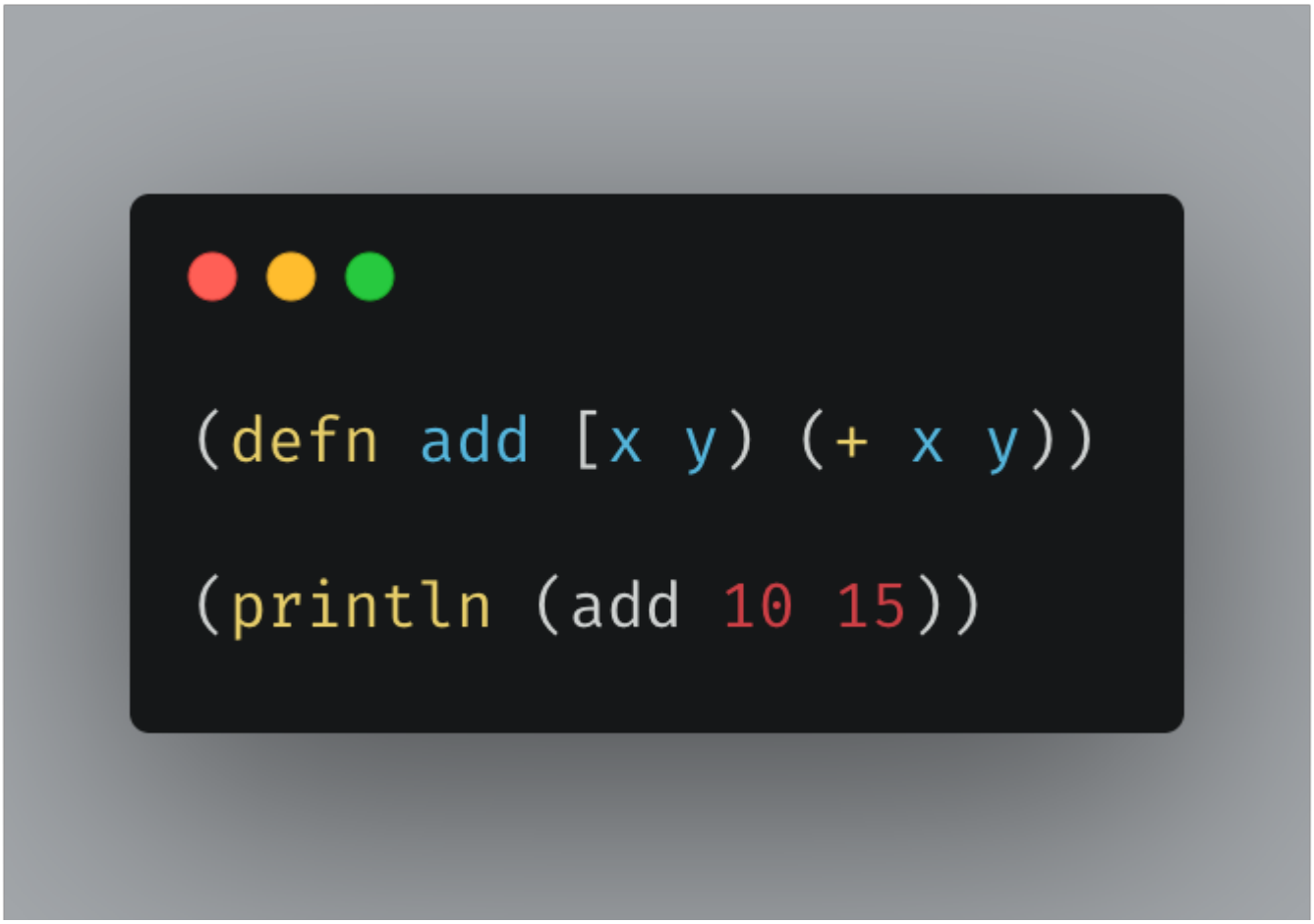
Twitter Thread by Ahmed Elhanafy

Ahmed Elhanafy
@ahmedlhanafy

I started learning Clojure in 2021 and there are a lot of things that I like about this language/ecosystem. I will try to summarize my experience in this thread.

If you're not familiar with Clojure, here is how the code looks like. Clojure is a Lisp-derivative which is different than most Fortran/C-derivative languages that we use today.



At first, the syntax was very off-putting (my god these parentheses!!), but then you start appreciating the elegance of the language and after sometime you stop being bothered by the syntax.

One of the things that Clojurists (I guess that's what they call themselves ■) always talk about is how they think of everything as data even the language's syntax! Every Clojure expression is just a plain ol' Clojure list that the complier reads

and executes.

And for that reason macros in Clojure are first class citizens (in fact a lot of Clojure built-in functionality is built with macros). Macros are functions that take your code as input (remember your code is just a list data structure) and it transforms your code into...

...another data structure that can be fed into the compiler, and with that you can invent your own syntax and build your own compiler tools. This alone gives you a lot of power. I will give a concrete example of using macros in a later tweet.

Clojure comes with a VERY rich standard library, think of Lodash built into the language, you will find whatever you fancy in that standard library.

Clojure also has novels tools. Two tools which helped my tremendously while writing Clojure code are Parinfer (@shaunlebron) and proto-repl (@jasongilman). Parinfer is an editor plugin that lets you focus on writing Clojure code without having to mess around parentheses...

you just write the opening parenthesis and the tool infers the closing one based on indentation. This tool is brilliant and it helped me avoid many silly mistakes. Check out the website: https://t.co/hDXhCGf3OA

```
(defn component []
  (html)
  [:div {:style {:background "#FFF"}
         :color "#000"}]
  [:h1 "title"])
```

Proto-repl, on the other hand, is a Clojure development environment in Atom, which allows you to execute your program inside the editor and see the results immediately. This also helped a lot while learning. I highly recommend you check it out; it works like magic ■.

One thing that was off-putting coming from writing Typescript daily is the lack of types and therefore all the tooling that benefits from that. Clojure is a dynamic language which is mostly a great thing except for tooling IMO. Code completion is far from perfect in all major...

...editors and the fact that the editor doesn't understand/infer your program makes it difficult to catch obvious bugs before running the code. I haven't tried clojure.spec and I don't know if it interops with editors/IDEs or not.

One important fact is that Clojure compiles to the JVM, but the amazing Clojure team also built ClojureScript which compiles to JS and you can use it to build SPAs (single page applications).

In my initial testing Clojure experience beats ClojureScript's, especially when it comes to error messages. Error messages in ClojureScript are so bad and I hope that the team fixes this because it's such a showstopper for beginners trying to learn the language for the first time

I was surprised to find that @RoamResearch and @Pitch (which are both quality products) are both built using ClojureScript, I hope that folks from these teams share their experiences with the language.

Now that you know that ClojureScript exists, I want to give an example of a brilliant macro that @roman01la wrote for writing css-in-js (or css-in-cljs I guess ■). cljss is library that allows you to write style definitions in ClojureScript and using Clojure macros...



... it compiles these definitions into class names and injects the styles in your document, all happening at build time. No need to fiddle around with Babel and ASTs, it all happens natively in the language!

There are tons of learning material out there but I attribute most of my learnings to "Clojure for the Brave and True" the amazing book written by the brilliant and humorous @nonrecursive. Also @richhickey the creator of the language has tons of great talks on YouTube...

here are the ones that I really liked:
- Simplicity Matters
- Clojure Made Simple
- Clojure for Java Programmers Part 1 & 2

@threadreaderapp unroll