

Implementing TULI5 in Elixir

Astrid Yu

CSC 430

2022-03-11

Elixir in a nutshell

- Functional, dynamically-typed, general-purpose language. A lot like untyped Racket.
- Released in 2012
- Comes with a REPL (`iex`)
- Lots of interop with Erlang
 - It's made in Erlang
 - Compiles to Erlang VM bytecode



elixir

StackOverflow Survey 2021

- Elixir pays the **3rd** most! (\$80,077) Tied with Erlang, likely because of how connected they are
- However, it is the **30th** most used language
- It is the **4th** most loved language (72% love vs 28% dread)

These statistics seem similar to lots of other functional programming languages

Language features

- Standard stuff: numbers, strings, variables, closures, tuples...
- Everything is an expression
- Can be compiled (.ex) OR interpreted (.exs)
- Maps (basically dictionaries)

Probably stolen from Lisp

- “Atoms,” which are just symbols by another name
 - Elixir's `:abc` is Racket's `'abc`
- Linked lists
- AST-based macros, along with a certain `quote/1` and `unquote/1`

Constructing TULI in Elixir

```
[ :fn, [ :x ], [ :+, [ :-, :x, 9 ], 10 ] ]
```

What I've made

- A working interpreter and parser
- I didn't bother to make most of the builtin environment
- Unknown how well it deals with edge cases, but oh well

Demo

Would I take a job writing Elixir?

- Maybe, it seems kinda interesting
- It does seem to have nice tooling
- But also, I don't really like dynamically typed languages

Thanks for listening!

Any questions?

Astrid Yu

Website: astrid.tech

Email: astrid@astrid.tech

Github: [@astralbijection](https://github.com/astralbijection)