



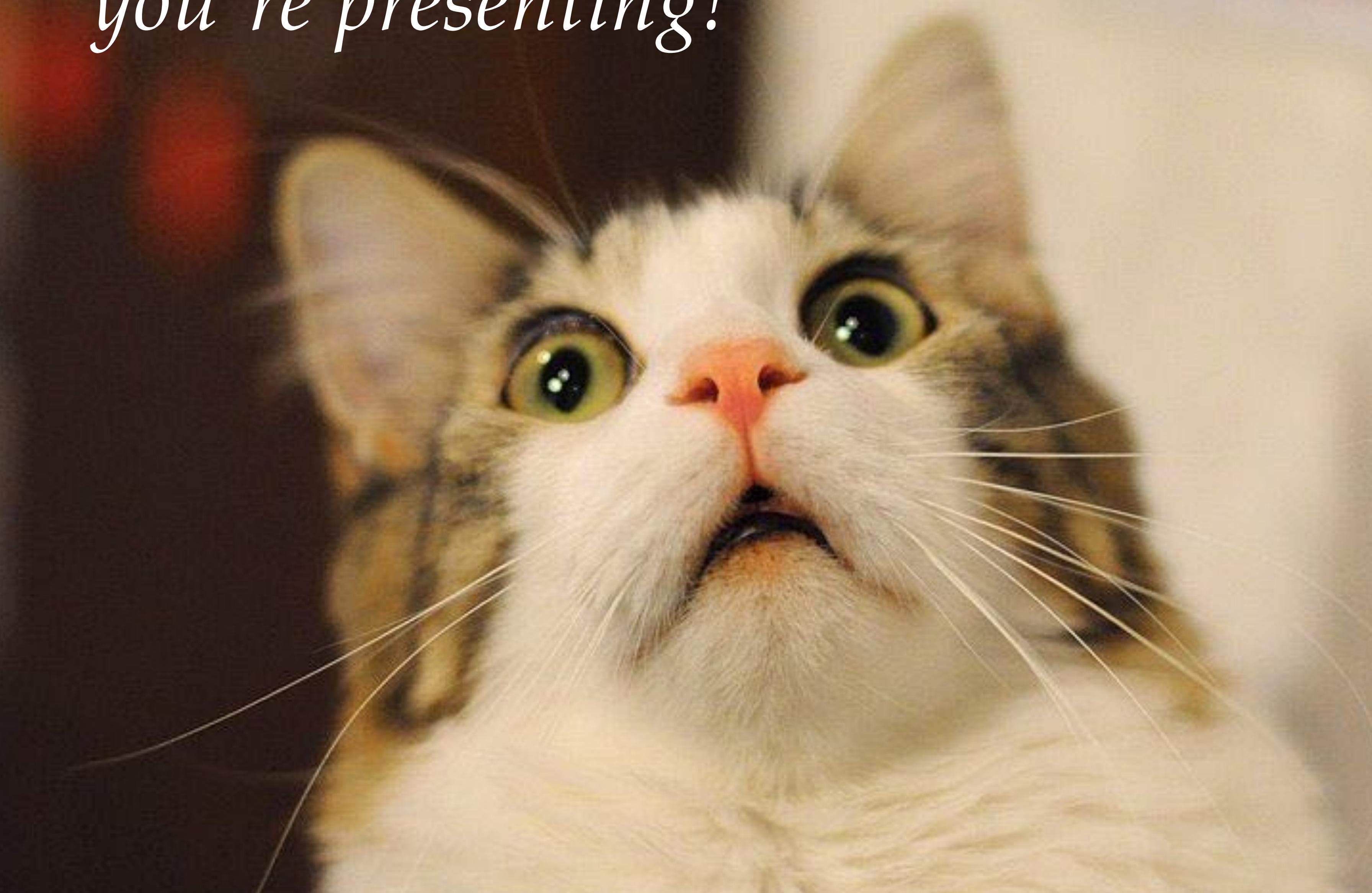
*Fullstack Academy Lightning Talk*

---

# Monadic Parser Combinators

a demo of functional benefits

*Surprise,  
you're presenting!*

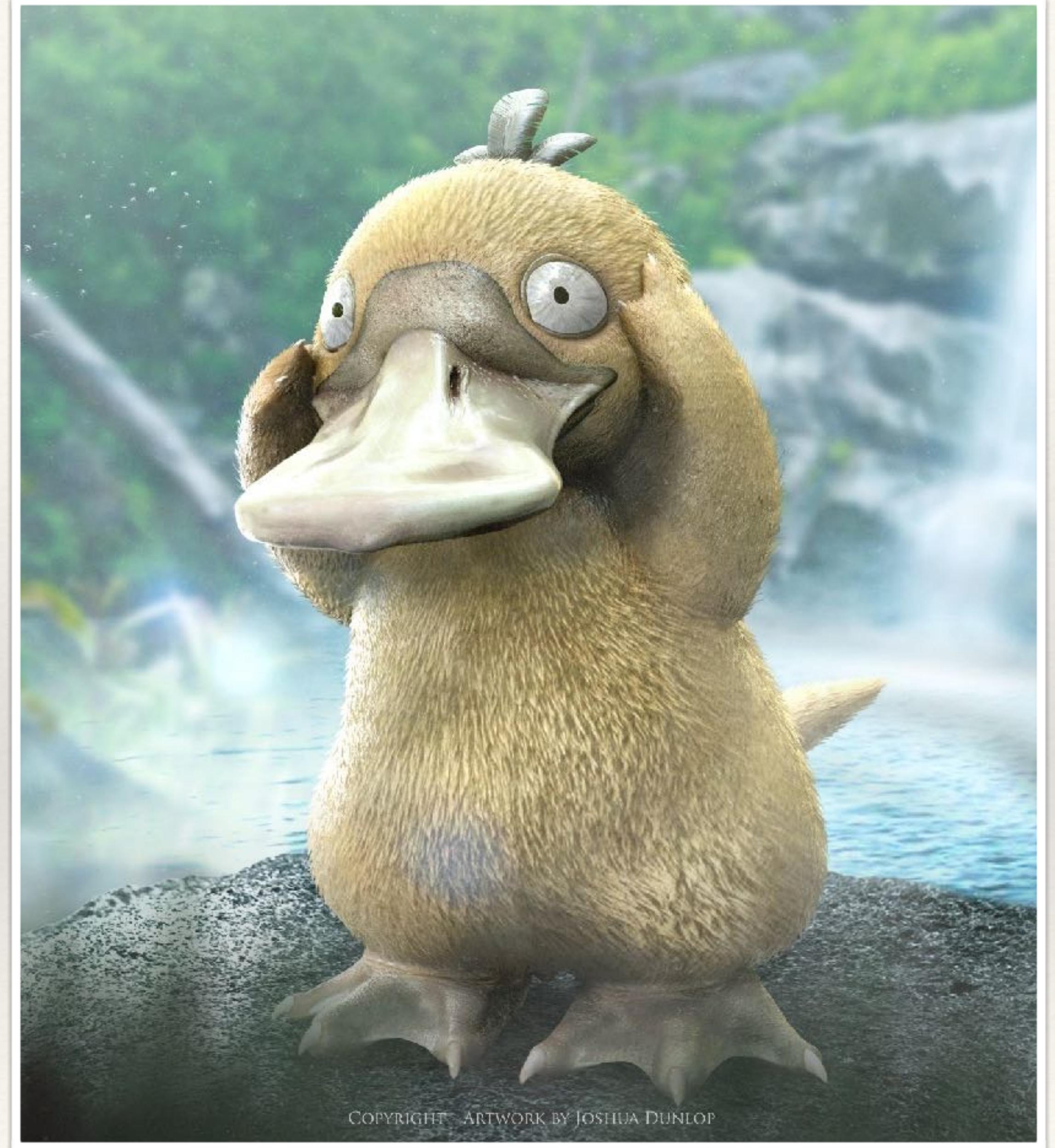


---

## *A Wild Topic Appears*

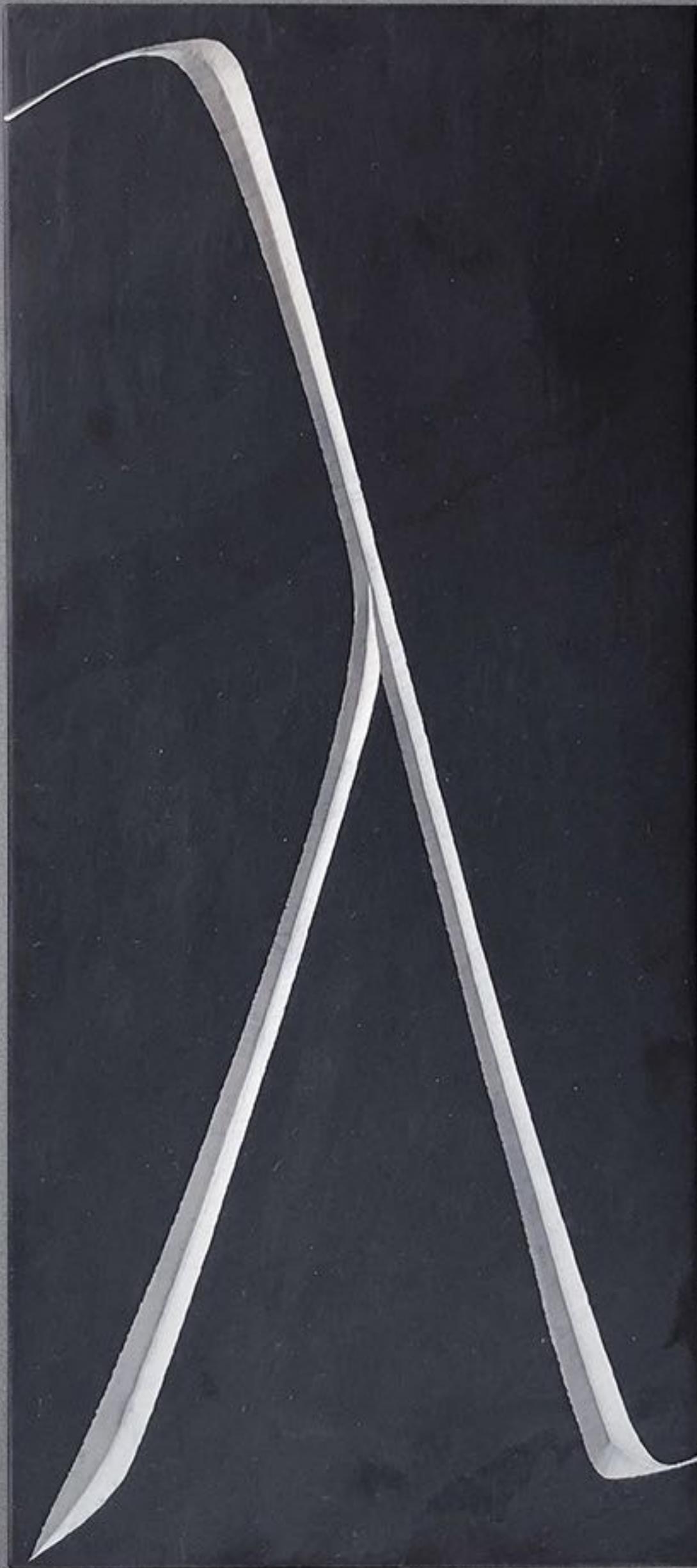
---

- Futures / Tasks?
- Folktale / Sanctuary / FantasyLand?
- Sum Types / Algebraic Data Types?
- Recursion Schemes?



COPYRIGHT: ARTWORK BY JOSHUA DUNLOP

I



*(for real though)*



*Fullstack Academy Lightning Talk*

---

# Monadic Parser Combinators

I choose you!

---

# What Do We Parse?

---

- ❖ JS -> abstract syntax tree / executable
- ❖ JSON / YAML -> objects
- ❖ (X/HT)ML -> DOM
- ❖ SEMVER -> structured data
- ❖ Math -> tree / number
- ❖ Phone numbers -> country, area, etc.
- ❖ CSV -> array of objects
- ❖ *Anything...* if it has a formal grammar

## Dr. Seuss on Parser Monads:

```
type Parser a = String → [(a, String)]
```



A Parser for Things  
is a function from Strings  
to Lists of Pairs  
of Things and Strings!

# Slightly Adapted for JS

---

Parser a :: String -> { result: a, remaining: String } | Null

parseGreeting("Hi!!") -> { result: "Hi", remaining: "!!" }

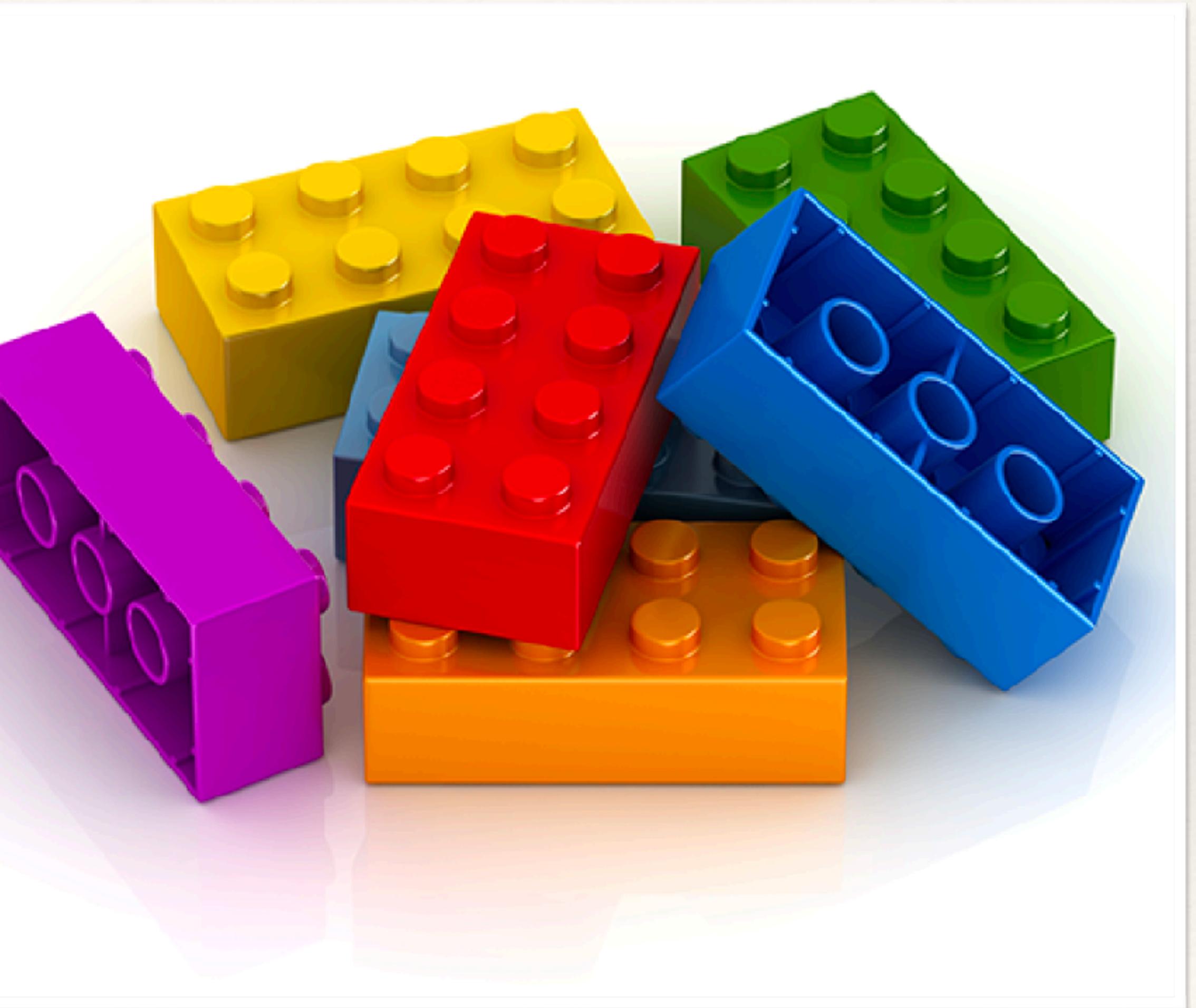
parseGreeting("Yawn") -> null

parseDigitAsInt("456") -> { result: 4, remaining: "56" }

parseDigitAsInt("hi") -> null

# What & Why?

- ❖ **Parsers**
  - ❖ Functions which analyze linear input into structured data
- ❖ **Combinators**
  - ❖ Remix simple parsers into richer versions
- ❖ **Monads**
  - ❖ *OH THE HUMANITY!*
  - ❖ ...can un-nest parsers in a smart way



D E M O