

10,000 FOOT OVERVIEW

CLOJURE

WHAT IS CLOJURE?

- ▶ Dialect of LISP
- ▶ Dynamic
- ▶ Designed to be hosted
 - ▶ JVM
 - ▶ JavaScript
 - ▶ CLR
- ▶ Pragmatic

WHY CLOJURE?

- ▶ Polyglot of Technology
 - ▶ Open to experimentation (i.e. crazy co-worker made us try it!)
- ▶ Faster Delivery / Maintainable Code
 - ▶ Processing Data
 - ▶ Building applications with simpler tools
- ▶ Rise of JavaScript (along with SPA's)
 - ▶ Code Sharing
 - ▶ Target Multiple Platforms
- ▶ Clojure is fun!

WHO ELSE USES CLOJURE?

- ▶ WalMart (WalMart Labs Division)
 - ▶ Processes every receipt through the store
- ▶ Netflix
 - ▶ MQL (Mantis Query Language) 2 Trillion Events / Day
- ▶ Amazon, Apple, CircleCI
- ▶ <https://clojure.org/community/companies>

A little bit of syntax...

PRIMITIVES & DATA STRUCTURES

```
;; Numerics
42                ; 64 Bit Integer
42N               ; BigInt - Arbitrary Precision
22/7              ; Ratio

;; Character
"Hello World!"    ; String
\e                ; Characters

;; Others
:user             ; Keyword
:user/first-name  ; Namespaced Keyword
map               ; Symbol
clojure.core/map  ; Namespaced Symbol

;; Data Structures
'(1 2 3 4)        ; List
[1 2 3 4]          ; Vector
#{1 2 3 4}         ; Set
{:a 1 :b 2 :c 3}   ; Map
```

PROGRAM STRUCTURE

```
(ns lunchlearn.core
  (:require [clojure.string :as str]
             [insights.core :as insights]))

(def a 42)

(def my-func
  (fn [name]
    (println "Hello," name)))

(defn my-func2
  "Given a name, it will print the greeting
   'Hello, {name}' to stdout"
  [name]
  (println "Hello," name))

(my-func2 "John")
```

(CODE-DEMOS)

RESOURCES FOR LEARNING MORE

<https://www.infoq.com/presentations/Simple-Made-Easy>

<https://clojure.org/>

<https://clojurescript.org/>

<https://www.braveclojure.com/foreword/>

<https://aphyr.com/tags/Clojure-from-the-ground-up>