Introduction

- Aaron Crow from Factual
- I'm mainly a Java developer
- I really like Clojure
  - Lisp on the (awesome!) JVM
  - Functional
  - Interesting strengths
- Factual is using Clojure more and more

“some real work with Clojure”

- Simple, fast talk
- Show bits and pieces of Clojure programming
- A little real work with simple data structures

Anonymous fns (“closures”)

```
(fn [x] (println "my arg is" x))
```

OR...

```
#(println “my arg is” %)
```

Anonymous fn example

```
> (def f #(println "my arg is" %))
#'user/f
> (f 88)
my arg is 88
> (f "factual.com")
my arg is factual.com
```

Clojure's map

- (map f coll)
- **f** is a first class function. Takes a thing, returns a thing
- **coll** is a collection of things
- **map** returns a new, transformed collection

map examples

```
; Bump up some integers:

(map inc [0 1 2 3 4 5 6])
=> (1 2 3 4 5 6 7)


; Get the squares of some integers:

(map #(* % %) [0 1 2 3 4 5 6])
=> (0 1 4 9 16 25 36)
```

Clojure and JSON

```
> (read-json "[{\"meetup\": \"SMJUG\"}]")
[{:meetup "SMJUG"}]


> (def mymaps
   [{:meetup "SMJUG" :city "Santa Monica"}
    {:meetup "Clojure" :city "Century City"}
    {:meetup "Scala" :city "Century City"}
    ])
> (get (first mymaps) :meetup)
"SMJUG"
```

Quick detour: get-in

"It is better to have 100 functions operate on one data abstraction than 10 functions on 10 data structures."

```
> (def mymap
  {:levelA {:levelB {:mykey "myval"}}})

>(get-in mymap [:levelA :levelB :mykey])
"myval"
```

Putting it together, using Factual's Places data

```
[
{:status "1",
  :country "US",
  :longitude -118.474,
  :factual_id "db71917f-2a7f-4e1b-be86-24c4d1844e28",
  :name "Yahoo",
  :postcode "90404",
  :locality "Santa Monica",
  :latitude 34.0307,
  :region "CA",
  :address "2450 Broadway",
  :tel "(310) 315-1870",
  :category
  "Business & Professional Services > Equipment, Supplies & Services > Telecommunication Services",
  :address_extended "Ste 600"}
{...}
...]
```

Transform a collection (Clojure)

```
(map :name places)
```

```
("Brown Barry H Attorney At Law"
 "Pfizer Health Solutions"
 "Danjaq"
 "Onewest Bank Fsb"
 "La Art Exchange"
 "Yahoo"
 "Lasik Vision"
 "Washington Post Advertising"
 "Rmc Water and Environment"
 "Helen's Cycles"
 ...
)
```

Transform a Collection (Java)

```
// Uses Google's handy Guava library

private Collection placeNames(Collection
places) {
  return Collections2.transform(places,
    new Function<JSONObject, String>(){
      @Override
      public String apply(JSONObject place) {
        return place.getString("name");
      }
    }
  );
}
```

Aaron's Shameless Plugs

1) Blog posts about Clojure, for us Java developers:

http://blog.factual.com/

2) Upcoming talk at LA Hacker News meetup:

“Is Clojure the best way to wrap JSON-based web APIs?”
Coloft, Saturday, October 1, 2011, 3:00 PM

Java Interop

```
(String. “a new string”)

(.length my-str)

(doto (java.util.HashMap.)
  (.put "a" 1)
  (.put "b" 2))

;; but might miss the point
```

What about Types?

```
(defn length-of [^String s]
  (.length s))
```