Clojure

How does it stack up?

Why do people use this language?

- Web scraping
- Data processing
- Concurrency
- Functional
- It's a Lisp
- JVM Language
- Web framework

Web Scraping

- Python (BeautifulSoup)
  - Python is more readable and writable
  - Python is faster
- Javascript (Node)
  - JS has jQuery for element selection
  - Node is faster

Data Processing

- Python
  - Python is faster
  - Python is more readable and writable
  - Has Matplotlib and Numpy
- R
  - R is faster
  - RStudio has built in graphics support
  - R is more readable and writable
- Matlab
  - Matlab is more user friendly
  - Matlab is faster then Clojure

Concurrency

- Go
  - Clojure is slower
  - Syntax is similar for channels
  - Go has inferred static typing (can pass channels as read-only or write-only)
  - Clojure uses transactions to mutate data which makes it safer
  - If you program using CSP, Go is just as safe
- Rust
  - Clojure is slower
  - Rust catches errors at compile time
  - Rust has inferred static typing
  - Rust separates channel readers and writers
  - The Rust compiler will not allow data races and is very safe

Functional Simplicity

- Haskell
  - Clojure is syntactically larger
  - Haskell has automatic currying
  - Haskell has parameter pattern matching
- Elixir
  - JVM is faster then BEAM
  - Lein is slower than Mix
  - Elixir syntax is better
  - Elixir has Actor model built in

Lisp’s and Metaprogramming

- Common Lisp
  - Clojure is restricted by the JVM
  - No first class access to continuations
  - No CPS transformation

Metaprogramming

Imagine a guy who builds cars. Say it's the same thing as using a computer. At some point he realizes he's always doing the same thing, more or less. So he builds factories to build cars, and it's much better. He's now programming! Nevertheless, once again, at some point, he realizes he's always doing the same thing, to some extent. Now he decides to build factories that build factories that build cars. That's metaprogramming.

JVM languages

- Kotlin
  - Kotlin is more similar to Java
  - Better currying syntax
  - Inferred static typing
  - Gradle and Maven are better then Leiningen
  - Clojure is safer without using CSP in Kotlin
  - Clojure is not as convenient with Spring
- Scala
  - Scala is more similar to Java
  - Better currying syntax
  - Inferred static typing
  - Sbt is better than Leiningen
  - Clojure is safer without using Actor model in Scala

Web Framework (ClojureScript)

- Javascript
  - Javascript can integrate server code with browser code more easily
  - Vue, Angular, React have better communities for web development
- Php
  - Laravel, Symfony have better communities for web development
- Ruby
  - Ruby on Rails, Sinatra are simpler and have larger communities
  - Ruby gives first class access to continuations

Personal Gripes and Biases

- Error messages give the print the full Java stack-trace
- Limited IDE support
- Lisp style languages are confusing to read
- I prefer static typing for complex projects

Summary

- This language does a lot of things OKly
- It does not have a single defining feature that you should use it for
- It has good community support
- You should use this language if
  - You love Lisp style languages
  - You want portability (JVM)
  - You want homoiconicity
  - You want to use a good Minikanren distribution