Simple

- Data-oriented
  - string, number, boolean, keyword, character, nil
  - (), {}, #{}, [] ;The basic collections
- (ᗡ • • •) ;Calls are functions, special forms, or macros followed by arguments
- Simple doesn’t necessarily mean easy
  - With the right resources, getting started is very fast

https://clojure.org/reference/evaluation

Data Oriented

- Clojure programs are written in the basic literal data structures of the Clojure programming languages
    - {}, [], #{}, ()
- The preferred way to encode domain models is as data, not classes (but classes are fully supported)
- Clojure has powerful data abstractions that extend across the language for reuse
- Clojure is Homoiconic, meaning it is written in its own data structures
    - The language can be extended at will using macros
    - Despite being a powerful feature, you generally don’t write many macros
- Clojure is easy to read, reason about, and debug
- There are no other mainstream JVM (or other?) languages that are as data-oriented as Clojure