Threading Macros

- Clojure forms naturally nest, evaluating inside out
- We naturally read left-to-right or top-to-bottom
  - Most cultures
  - Nobody reads inside-out
- Threading macros reverse the nesting order
  - Easier to read
  - “Natural” computational pipelines
  - They are macros - Code is actually rearranged

The Threading Macro Zoo

- Number of arrowheads denotes threading position
  - -> First position
  - ->> Last position
  - The macros
    - ->, ->>, Simple thread first and thread last
    - some->, some->>, Act as a short circuit. When any form returns nil, the macro returns nil
    - cond->, cond->>, Conditionally evaluate a form based on a precondition
    - as->, as->>
  - https://clojure.org/guides/threading_macros