

# as->, as->>

- Use when you want to bind a symbol through your pipeline
- Use when both -> and ->> are needed or are becoming awkward
- I don't see this macro used a lot
- Often times an intermediate result in a let form or a separate function solves the same issue more cleanly

```
(as-> [:foo :bar] v  
      (map name v)  
      (first v)  
      (.substring v 1))
```

# Threading Macros

- Rearrange code to make it more readable
- Use macroexpand to see what is happening
- Creates natural pipelines
- ->> and ->> are most common, others are used as needed
- See [https://clojure.org/guides/threading\\_macros](https://clojure.org/guides/threading_macros)