




**Erik Assum** @slipset · 3h

Working on my [@flatmaposlo](#) talk about transducers in [#Clojure](#), I discovered that while ``max`` and ``min`` “works” with reduce, while they’re not proper reducing fns (as required by Clojure’s reduce), they do form monoids together with numbers. :/

 1   1 

**Erik Assum** @slipset · 1h

Would be nice with a fn like





```
(def reducing [f identity]
  (fn
    ([] identity)
    ([x y] (f x y))))
```


for reducers akin to ``completing`` for transducers

Then you could do something like

```
(reduce (reducing max Integer/MIN_VALUE) numbers)
```

safely

 1   

**Jean Niklas L'Orange** @hyPiRion


Following





Replying to [@slipset](#) [@flatmaposlo](#)

There is! `clojure.core.reducers/monoid` :)

3:38 PM - 27 Apr 2019

1 Retweet 1 Like



  1  1 

(reduced x)