

readkinetic.com | @read_kinetic



REALKINETIC

YOUR POTENTIAL REALIZED


```
-- This is a fold left of Ints
-- The left means we reduce (or traverse) from the left
foldlInt : (Int -> List Int -> List Int) -> Int -> List Int -> Int
foldlInt func aggVal list =
    case list of
        [] ->
            aggVal

        x :: xs ->
            foldl func (func x aggVal) xs
```

We'll start with a "fold"

(AKA: Reduce)