Reduce

eg:-   let arr= [2,3,4,5,6]

(Sum of all elements) → 2+3+4+5+6

↓

(20) uc.

Approach

↳ let sum=0,

$sum= 2$

$sum= 2+3$

$sum= 2+3+4$

$sum= 2+3+4+5$

$sum= 2+3+4+5+6$

let sum= arr. reduce (function (accumulator, value)
&

}, 0)

(∵ we are defining sum=0, in
reduce we initialize
here)

(let sum=0)

Accumulator

↓
bcz it accumulates value .

let sum = arr. reduce(function(sum, value) {

let updatedsum = sum + value

return updatedsum

}, 0).

Dry Run

let arr = (2, 3, 4, 5, 6) ✓.   Date ....

let sum = arr.reduce (function (sum, value) { ①

     let updated sum = sum + value; ②
     return updated sum → reduce returns only
                    one time after executing
     }, 0);   whole prog.
          2 5 9 14

① sum = 0,   ② = value

     updated sum = 0 + 2 = $\boxed{2}$.

② value = ③ ✓

     updated sum = 2 + 3 = $\boxed{5}$.

③ value = ④ ✓

     updated sum = 5 + 4 = $\boxed{9}$.

④ value = ⑤ ✓

     updated sum = 9 + 5 = $\boxed{14}$

⑤ value = ⑥ ✓

     updated sum = 14 + 6 = $\boxed{20}$ ✓.

               $\boxed{20}$ ✓

Spiral