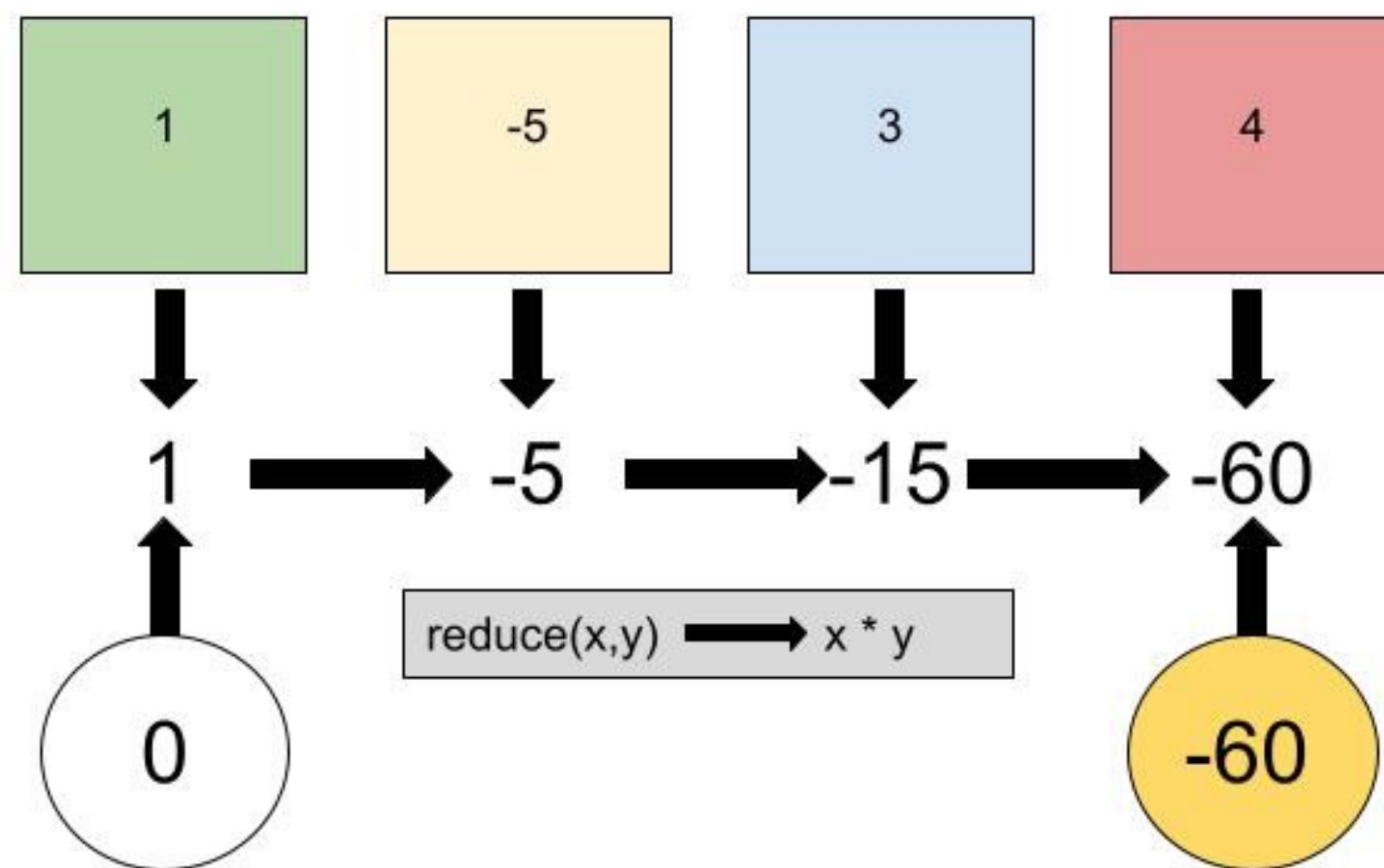




REDUCE – EJEMPLO – MULTIPLICADOR

- ▶ Vamos a obtener la multiplicación de todos los valores de la lista.
- ▶ Individualmente irá obteniendo el valor y añadiéndolo al acumulador.



```
from functools import reduce
values = [1, -5, 3, 4] # -60
def multiply_all(total = 0, value = 0): return total * value

reduce(multiply_all, values) # -60

reduce(lambda x, y : x * y, values) # -60
```



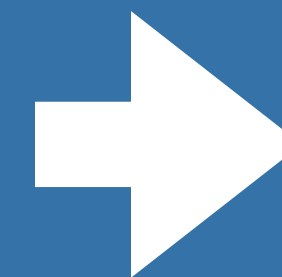
RECURSIVIDAD

- ▶ Técnica muy utilizada en programación.
- ▶ Nos permite ejecutar un bloque de instrucciones un cierto número de veces (lo especificado).
- ▶ Funcionamiento similar a los bucles, pero en este caso llamando una función a si misma.

```
# Sin recursividad
for i in range(10, 0, -1):
    print(i)

# Recursividad
def discount_values(values):
    if len(values) == 0:
        return
    print(values[0])
    return discount_values(values[1:])

print(discount_values(list(range(10, 0, -1))))
```



```
10
9
8
7
6
5
4
3
2
1
10
9
8
7
6
5
4
3
2
1
```